UNIVERSIDADE ESTADUAL DO MATO GROSSO DO SUL
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO- PROPP
Departamento de Matemática
MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL - PROFMAT

EWERTON DA SILVA SCHROEDER

CÓDIGOS BINÁRIOS E TRUQUES DE MÁGICA

DISSERTAÇÃO DE MESTRADO

DOURADOS - MS
2017

UNIVERSIDADE ESTADUAL DO MATO GROSSO DO SUL
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO- PROPP
Departamento de Matemática
MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL - PROFMAT

EWERTON DA SILVA SCHROEDER

CÓDIGOS BINÁRIOS E TRUQUES DE MÁGICA

Dissertação submetida como requisito parcial para obtenção do grau de Mestre, pelo Curso de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT junto PRÓ-REITORIA DE PESQUISA E PÓSGRADUAÇÃO- PROPP da Universidade Estadual do Mato Grosso do Sul.
Orientador: Prof. Dr Otávio José Neto Tinoco Neves dos Santos

DOURADOS - MS
2017

S395c Schroeder, Ewerton da Silva
Códigos binários e truques de mágica / Ewerton da Silva Schroeder. – Dourados, MS: UEMS, 2017.
75 f. : il. ; 30cm.

Dissertação (Mestrado) – Universidade Estadual de Mato Grosso do Sul, Curso de Mestrado Profissional em Matemática em Rede Nacional, 2017.
Orientador: Prof. Dr. Otávio José Neto Tinoco Neves dos Santos.

1. Códigos 2. Truques 3. Teoria dos códigos 4. Códigos de Hamming I. Título.

CDD 23.ed. 512

UNIVERSIDADE ESTADUAL DE MATO GROSSO DO SUL
PROGRAMA DE MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL - PROFMAT

EWERTON DA SILVA SCHROEDER

CÓDIGOS BINÁRIOS E TRUQUES DE MÁGICA

Produto Final do Curso de Mestrado Profissional apresentado ao Programa de Pós-Graduação *Stricto Sensu* em Matemática em Rede Nacional, da Universidade Estadual de Mato Grosso do Sul, como requisito final para a obtenção do Título de Mestre em Matemática.

Aprovado em: 02/06/2017

BANCA EXAMINADORA:

Prof. Dr. Otávio José Neto Tinoco Neves dos Santos (UEMS)
Universidade Estadual de Mato Grosso do Sul

Prof. Dr. Cosme Eustáquio Rúbio Mercedes (UEMS)
Universidade Estadual de Mato Grosso do Sul

Prof. Dra. Irene Magalhães Craveiro (UFGD)
Universidade Federal da Grande Dourados

Dedico esse trabalho primeiramente a Deus pois
"Ele é meu refúgio e fortaleza, socorro bem presente na hora da angústia" Sl. 46:1
À minha esposa Daiane que esteve a meu lado me apoiando e encorajando a continuar.
A meus pais Edson e Elisabete que me proporcionaram uma vida de dedicação ao estudo.

Tal é a confiança que temos diante de Deus, por meio de Cristo.
Não que sejamos capazes, por nós mesmos,
de pensar alguma coisa, como de nós mesmos;
mas a nossa capacidade vem de Deus.

II Coríntios: 3: 4,5; Bíblia Sagrada

Agradecimentos

Agradeço primeiramente a Deus, pois "a Sua palavra é lâmpada que ilumina os meus passos e luz que clareia o meu caminho" Sl: 119:105; e estendo os agradecimentos a todos os que contribuíram com a realização desse trabalho, especialmente:

- à minha esposa, Daiane, por suas orações, pelo apoio, companheirismo, compreensão, paciência e força em todos os momentos;

- ao meu pai, Edson, que não teve oportunidade de estudar, mas os seus sonhos estenderam-se a mim através de palavras de conforto e encorajamento. Meu êxito será o dele também;

- a minha mãe, Elisabete, que sempre me fez lembrar que a nossa jornada não acaba onde chegamos hoje.

- ao Prof. Dr. Otávio, meu orientador. Agradeço pela escolha do tema, pelo apoio, estímulo e pelas dicas importantes no desenvolvimento desse trabalho, e ainda por suas horas de dedicação a ajudar;

- aos professores do PROFMAT, da UEMS, pela partilha do conhecimento através das disciplinas oferecidas no decorrer do curso;

- aos amigos do PROFMAT, pelas inúmeras contribuições durante os estudos e pelos longos períodos que passamos nos dedicando ao curso.

- à CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior pelo apoio financeiro, sem o qual não haveria a possibilidade de conclusão desse trabalho.

Resumo

Em Matemática, a Álgebra é o campo que gera maior aversão aos alunos da educação básica e é vista como um dos seus campos mais complexos sendo que ela move-se quase exclusivamente no campo dos conceitos abstratos e de suas inter-relações. A vitalidade da Matemática deve-se ao fato de que, apesar de seu caráter abstrato, seus conceitos e resultados têm origem no mundo real e encontram muitas aplicações em outras ciências e em inúmeros aspectos práticos da vida diária. Neste trabalho mostraremos a Teoria dos Códigos Detectores e Corretores de Erros, apresentamos os conceitos de Números Binários, Matriz Binária, Matrizes de Código e o Código de Hamming e algumas aplicações a conteúdos do Ensino Fundamental e Ensino Médio em formato de Truques de Mágica, formando um elo entre teoria e prática.

Palavras-chave: Códigos, Truques, Mágica, Teoria dos Códigos, Hamming.

Abstract

In Mathematics, Algebra is the field that generates greater aversion to the students of the basic education and is seen like one of its more complex fields being that it moves almost exclusively in the field of the abstract concepts and their interrelationships. The vitality of Mathematics is due to the fact that, despite its abstract character, its concepts and results originate in the real world and find many applications in other sciences and in many practical aspects of daily life. In this work we will show The Theory of Error-Correcting and Detection Codes, we present the concepts of Binary Numbers, Binary Matrix, Code Matrices and the Hamming Code and some applications to Elementary and High School contents in Magic Tricks format, forming a link between theory and practice.

Keywords: Codes, Tricks, Magic, Codes Theory, Hamming.

Sumário

Introdução

A Teoria dos Códigos teve início nos anos de 1940, nesta época, os computadores custavam alguns milhões de dólares e seu uso estava restrito a grandes instituições. O Laboratório Bell, uma instituição de pesquisa tecnológica localizado nos EUA, possuia estes computadores, nesse período a programação era feita em cartões perfurados, caso houvesse um erro na perfuração do cartão o computador interrompia a leitura e passava para o próximo programa. Richard W. Hamming trabalhava no Laboratório Bell em 1947, mas ele tinha acesso ao computador apenas nos fins de semana.

De acordo com [11] Richard Hamming perdeu dois finais de semanas consecutivos, porque deixara seus programas para serem executados, mas por erro na programação (perfuração dos cartões), nada tinha sido feito. Depois disso ele pronunciou: *"Maldição, se as máquinas podem detectar um erro, porque não podem localizar a posição do erro e corrigi-lo?"*

Foi a partir dessa história que surgiram os códigos corretores de erros. Hamming desenvolveu um código capaz de detectar até dois erros e corrigir um erro, se ele for o único. Atualente a Teoria dos Códigos Corretores de Erros está diretamente ligada com a matemática, computação e engenharia elétrica.

As primeiras aplicações de Códigos Corretores de Erros foram desenvolvidas para sistemas espaciais e de comunicação por satélite da NASA. Quando a a nave espacial Voyager enviou fotos das luas de Júpiter em 1979, estava a 640 milhões de quilômetros da Terra. Um ano depois, enviou fotos dos anéis de Saturno. Em 1989, após ter passado 12 anos viajam pelo espaço e quase cinco bilhões de quilômetros da Terra, conseguiu transmitir imagens incrivelmente detalhadas e perfeitamente focadas de Tritão, a maior lua de Netuno. Se não houvesse um sistema de correção de erros todo esse investimentos e os anos de espera poderiam ter sido em vão.

Os Códigos Corretores de Erros também são amplamente utilizados em sistemas de armazenamento de dados, por exemplo: CDs, DVDs, Pendrive, etc. Um CD de áudio pode conter até 70 minutos de música, um segundo de música tem cerca de 1,5 milhão de bits. Esses bits são

representados por cavidades na superfície do CD, arranhões, poeira e impressões digitais causam erros que são eliminados usando códigos corretores de erros.

Os códigos corretores de erros também estão presentes na Comunicação móvel (telefonia celular, redes sem fio, etc), na transmissão por satélite (TV, rádio, etc). Todo sistema de transmissão digital de informações os códigos corretores de erros são empregados. Nesses sistemas podem ocorrer problemas que fazem com que a mensagem recebida seja diferente da mensagem enviada. O objetivo da Teoria dos Códigos é desenvolver sistemas que permitam detectar e corrigir eventuais erros na transmissão de informações digitais.

A situação geral considerada em Teoria de Códigos pode ser esquematizada na seguinte figura:

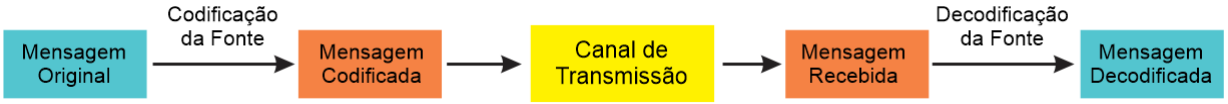


Figura 1: Esquema de codificação e decodificação

Para entendermos melhor os conceitos da Teoria dos Códigos vamos fazer uma analogia com um código muito utilizado por todos nós, o nosso idioma. Na língua portuguesa usamos um alfabeto de 26 letras e as palavras são sequências de letras. É claro que a língua não é composta por todas as "palavras" possíveis formadas a partir das letras, essa característica faz com que a língua portuguesa seja, em alguns casos, um código detector e corretor de erros. Suponhamos que, ao escrevermos uma palavra, tenhamos a sequência de letras "Guitaxra". Como este não é um elemento do nosso idioma, percebe-se logo que houve um erro; e como a palavra que mais se aproxima a "Guitaxra" é "Guitarra", a correção foi possível e fácil.

Vejamos um exemplo, baseado em [12], se pretendêssemos escrever a palavra "vaca", mas acidentalmente pressionou-se uma tecla errada e fosse escrita como "maca", ou como "faca", ou ainda "vaia", não detectaríamos o erro, dado que todas essas palavras fazem parte do nosso alfabeto. Com isso, vemos que esse código tem pouca eficiência pois nele existem palavras muito próximas uma das outras. Para aumentar sua eficiência podemos acrescentar dados adicionais ao codificar a palavra que será transmitida (chamados de redundâncias), e, por meio dessa informação extra, os dados podem ser recuperados. Voltando ao exemplo, codificando "vaca" como sendo "vaca leiteira", se houver um erro na palavra código poderemos agora decodificar corretamente a mensagem, dado que palavras como: "maca leiteira", "faca leiteira" e "vaia leiteira" não fazem sentido, e a palavra mais próxima é "vaca leiteira" que decodificando temos a palavra correta: "vaca". Mais uma vez, o contexto dar-nos-á a chave para encontrar a palavra certa.

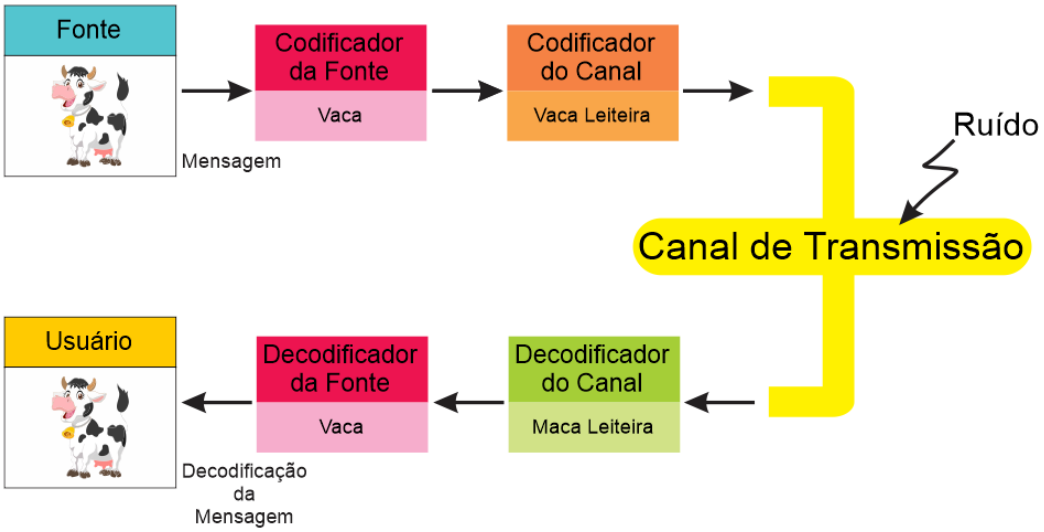


Figura 2: Esquema de codificação e decodificação

Os códigos corretores de erros baseiam-se fortemente na Álgebra e são excelente exemplo da aplicação Matemática. Ainda hoje, em Matemática, a Álgebra é o campo que gera maior aversão aos alunos da educação básica e é vista como um dos seus campos mais complexos. Tendo aulas com metodologias que enfatizam a memorização e repetição de exercícios de fixação.

Segundo os Parâmetros Curriculares Nacionais [2] "(...)A Matemática é componente importante na construção da cidadania, na medida em que a sociedade se utiliza, cada vez mais, de conhecimentos científicos e recursos tecnológicos. Além disso a atividade matemática escolar não é "olhar para coisas prontas e definitivas", mas a construção e a apropriação de um conhecimento pelo aluno. Para isso o professor deve utilizar recursos didáticos como jogos, livros, vídeos, calculadoras, computadores e outros materiais, pois eles têm um papel importante no processo de ensino e aprendizagem".

Sabemos que a Matemática move-se quase exclusivamente no campo dos conceitos abstratos e de suas inter-relações. Para demonstrar suas afirmações, o matemático emprega apenas raciocínios e cálculos. Mesmo em salas de aula do Ensino Fundamental e Ensino Médio, os resultados matemáticos são carregados de precisão e os raciocínios desenvolvem-se num alto grau de minuciosidade, que os torna incontestáveis e convincentes.

A vitalidade da Matemática deve-se ao fato de que, apesar de seu caráter abstrato, seus conceitos e resultados têm origem no mundo real e encontram muitas aplicações em outras ciências e em inúmeros aspectos práticos da vida diária: na indústria, no comércio e na área tecnológica. Quando o professor de matemática mostra aplicações da sua disciplina ela se torna mais interessante e, em consequência, os alunos aprendem mais. Códigos corretores de erros

baseiam-se fortemente na álgebra e são excelente exemplo da aplicação Matemática.

Nosso objetivo nesse trabalho é mostrar aplicações da Teoria dos Códigos Corretores de Erros a conteúdos do Ensino Fundamental e Ensino Médio, formando um elo entre teoria e prática.

Os pré-requisitos para ler este trabalho consistem em uma certa maturidade matemática, que é adquirida durante a Graduação em Matemática. Iniciaremos com a preparação ao estudo de códigos corretores de erros e veremos os conceitos básicos que o fundamentam, o que inclui: o sistema binário de numeração, o conjunto $\mathbb{Z}_2$ (conjunto números inteiros módulo 2), operações e propriedades das matrizes.

O capítulo 1 é dedicado ao sistema binário, iniciando com sistemas de numeração, passando pelo sistema posicional base 2, finalizando a primeira sessão com "Um truque com o Sistema Posicional na Base 2" e aplicável a alunos de séries iniciais. Em seguida ele trata sobre a detecção de erros e a paridade em matrizes de código binário e é finalizado com uma aplicação em "Um truque com Paridade".

No capítulo 2 é dedicado ao Código de Hamming, inicia com o conjunto $\mathbb{Z}_2$ (conjunto números inteiros módulo 2) e matrizes de código em $\mathbb{Z}_2$, apresentamos o Peso de Hamming, a Distância de Hamming e a Distância de Hamming Mínima, finalizando com o Código de Hamming e uma aplicação através de "Um Truque de Mágica Baseado no Código de Hamming".

Finalizamos o trabalho no capítulo 3, é apresentada uma proposta de roteiro para ministrar os conteúdos abordados no capítulo 1, contendo indicações metodológicas ao professor e atividades para os alunos, essas atividades serão divididas em duas, uma direcionada ao 9º Ano do Ensino Fundamental e continua com outra atividade com os conteúdos do capítulo 3 direcionada a alunos que estejam cursando, no mínimo, o segundo ano do ensino médio, uma vez que já terão visto o conteúdo de matrizes.

Finalmente nos anexos temos o material para ser impresso e utilizado em sala de aula, com uma proposta de aula baseada em códigos corretores de erros.

Capítulo 1

Códigos Binários

"Há apenas 10 tipos de pessoas:
as que entendem o sistema binário,
e as que não entendem."

Em [7] vemos que o método de numeração de quantidades ao qual estamos acostumados, tem dez valores possíveis {0, 1, 2, 3, 4, 5, 6, 7, 8, ou 9}, onde o símbolo 0 (zero) representa a ausência de algarismo. Por serem dez os algarismos, o sistema é chamado decimal. O sistema de numeração decimal é dito *posicional*, isto é, a posição que cada algarismos ocupa em um número altera seu valor, veja:

$$1985 \neq 9851$$

Os sistemas de numeração posicionais baseiam-se no seguinte teorema.

Teorema 1.1. Dados $a, b \in \mathbb{N}$, com $b > 1$, existem números naturais $r_0, r_1, r_2, \cdots, r_n < b$, univocamente determinados, tais que

$$a = r_n b^n + r_{n-1} b^{n-1} + \cdots + r_2 b^2 + r_1 b^1 + r_0 b^0$$

Demonstração. Vamos demonstrar o teorema usando o Princípio de Indução Completa sobre a.
Se $a = 0$, ou se $a = 1$, basta tomar $n = 0$ e $r_0 = a$.
Supondo o resultado válido para todo natural menor do que a, vamos prová-lo para a.
Pela divisão euclidiana, existem q e r únicos tais que

$$a = bq + r \text{ , com } r < b$$

Como $q < a$, pela hipótese de indução, segue-se que existem números inteiros $n' \geq 0$ e $0 \leq r_1, r_2, \cdots, r_{n'+1} < b$, com $r_{n'+1} \neq 0$, univocamente determinados, tais que

$$q = r_1 + r_2 b + \cdots + r_{n'+1} b^{n'}$$

Levando em conta as igualdades acima destacadas, temos que

$$a = bq + r = b(r_1 + r_2 b + \cdots + r_{n'+1} b^{n'}) + r$$

donde o resultado segue-se pondo $r_0 = r$, $n = n' + 1$. □

Essa representação dada no teorema é chamada de expansão relativa à base b. Quando $b = 10$, essa expansão é chamada *expansão decimal*, veja como fica para o número 1985.

$$1985 \quad = \quad 1 \times 10^3 + 9 \times 10^2 + 8 \times 10^1 + 5 \times 10^0$$

Comparando novamente os números 1985 e 9851 temos:

$$1985 = 1 \times 10^3 + 9 \times 10^2 + 8 \times 10^1 + 5 \times 10^0 \qquad \text{e} \qquad 9851 = 9 \times 10^3 + 8 \times 10^2 + 5 \times 10^1 + 1 \times 10^0$$

o que mostra que realmente são diferentes.

A base de um sistema de numeração é a quantidade de algarismos disponíveis na representação. A base dez é hoje a mais usualmente empregada, embora não seja a única utilizada. No comércio compramos ovos por dúzia (base doze) e marcamos o tempo em minutos e segundos (base sessenta), em especial trataremos aqui da base dois, a *base binária*, também chamada de *Sistema Posicional na Base 2*.

1.1 Sistema Posicional na Base 2

Vemos em [4] que o sistema binário, é um sistema de numeração posicional em que todas as quantidades são representadas utilizando apenas dois símbolos, os algarismos 0 e 1. Na computação e teoria da informação, *bit* é a sigla para *Binary Digit*, que em português significa dígito binário, esta é a menor unidade de informação que pode ser armazenada ou transmitida. Computadores descrevem dados em termos de 0s e 1s (que podem ser interpretados como desligado/ligado, fechado/aberto, falso/verdadeiro ou não/sim) e são idealizados para armazenar

instruções em múltiplos de bits, que são denominados *bytes* (um byte tem oito bits). A importância do Sistema Posicional na Base 2 se deve ao fato de tudo na informática ser medido através de bits e bytes.

Para evitar a confusão ao usar diferentes sistemas numéricos, a base de cada número individual será especificada subscrita para cada número, a menos que já seja subentendido qual base está sendo usada. Por exemplo, o número binário 1001 pode ser especificado como "base dois" escrevendo-o como 1001_2. O número 156 pode ser escrito como 156_{10} e lido como "cento e cinquenta e seis, base dez".

Observe os 5 primeiros números naturais na forma de potências de base 2.

$$\begin{aligned} 1 &= 1 \times 2^0 \\ 2 &= 1 \times 2^1 + 0 \times 2^0 \\ 3 &= 1 \times 2^1 + 1 \times 2^0 \\ 4 &= 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ 5 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \end{aligned}$$

Como queremos representar os números apenas usando 0s ou 1s usaremos cada um dos r_i de cada número natural, na posição que ele aparece. Assim temos:

$$\begin{aligned} 1_{10} &= 1_2 \\ 2_{10} &= 10_2 \\ 3_{10} &= 11_2 \\ 4_{10} &= 100_2 \\ 5_{10} &= 101_2 \end{aligned}$$

e assim por diante.

A conversão para a notação decimal de um número escrito na base 2 é feita usando o valor posicional de cada dígito.

Exemplo 1.2. Convertendo o número 1001101_2 para base decimal temos:

$$\begin{aligned} 1001101_2 &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 2^6 + 2^3 + 2^2 + 2^0 \\ &= 64 + 8 + 4 + 1 \\ &= 77_{10} \end{aligned}$$

Para converter um número da base decimal para a base binária, basta realizar divisões sucessivas por 2; os restos dessa divisão fornecerão os algarismos do número na nova base.

Exemplo 1.3. Vamos converter 75_{10} para base binária:

$$\begin{aligned} 75 &= 2 \times 37 + 1 \\ 37 &= 2 \times 18 + 1 \\ 18 &= 2 \times 9 + 0 \\ 9 &= 2 \times 4 + 1 \\ 4 &= 2 \times 2 + 0 \\ 2 &= 2 \times 1 + 0 \\ 1 &= 2 \times 0 + 1 \end{aligned}$$

Portanto, $75_{10} = 1001011_2$

Conferindo o resultado obtido, temos:

$$\begin{aligned} 1001011_2 &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 2^6 + 2^3 + 2^1 + 2^0 \\ &= 64 + 8 + 2 + 1 \\ &= 75_{10} \end{aligned}$$

1.1.1 Um Truque com o Sistema Posicional na Base 2

A Aritmética é uma excitante área da matemática, a qual usa técnicas e estruturas comuns do dia a dia para provar resultados mais abrangentes da matemática. Vejamos um truque de magia, baseado em [5], que ilustra de forma recreativa o alcance e a eficácia da Aritmética.

Esse truque pode ser aplicado de forma simples em sala de aula. Segue-o:

O mágico pede a um voluntário entre a turma que pense num número entre 1 e 15: Em seguida, pede ao voluntário que diga se o número em que pensou está em cada um dos cartões, se sim ou não.

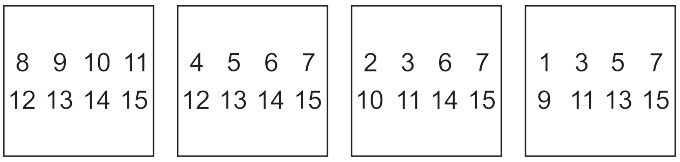


Figura 1.1: Truque com Cartões

Imediatamente o mágico adivinha o número. Veja o exemplo:

Exemplo 1.4. Supondo que o voluntário diga que o número pensado está no primeiro, no segundo e no quarto cartão, qual o número?

Resposta: O número 13.

Como adivinhou? Simplesmente somando o algarismo mais à esquerda da linha de cima em cada um dos cartões com resposta afirmativa. (No exemplo, $8 + 4 + 1 = 13$)

Como o truque acontece e porque funciona? Simples, este truque baseia-se na representação binária dos números naturais. Antes de explicá-lo vamos primeiramente à confecção dos cartões, parte fundamental do truque.

Escrevendo os números naturais de 1 a 15 em binários temos:

$$\begin{aligned}
1 &= 0001\\
2 &= 0010\\
3 &= 0011\\
4 &= 0100\\
5 &= 0101\\
6 &= 0110\\
7 &= 0111\\
8 &= 1000\\
9 &= 1001\\
10 &= 1010\\
11 &= 1011\\
12 &= 1011\\
13 &= 1101\\
14 &= 1110\\
15 &= 1111
\end{aligned}$$

Agora, trocando cada 1 e 0 da representação binária, por S (SIM) e N (NÃO), temos em quais cartões cada número deverá aparecer ou não:

$$\begin{aligned}
1 &= 0001 \rightarrow NNNS \\
2 &= 0010 \rightarrow NNSN \\
3 &= 0011 \rightarrow NNSS \\
4 &= 0100 \rightarrow NSNN \\
5 &= 0101 \rightarrow NSNS \\
6 &= 0110 \rightarrow NSSN \\
7 &= 0111 \rightarrow NSSS \\
8 &= 1000 \rightarrow SNNN \\
9 &= 1001 \rightarrow SNNS \\
10 &= 1010 \rightarrow SNSN \\
11 &= 1011 \rightarrow SNSS \\
12 &= 1100 \rightarrow SSNN \\
13 &= 1101 \rightarrow SSNS \\
14 &= 1110 \rightarrow SSSN \\
15 &= 1111 \rightarrow SSSS
\end{aligned}$$

Por exemplo, o número 11 em binário é $1011 \rightarrow SNSS$. O que mostra que 11 deve estar no 1^o, 3^o e 4^o cartão.

Segue assim a construção dos cartões inicialmente apresentados.

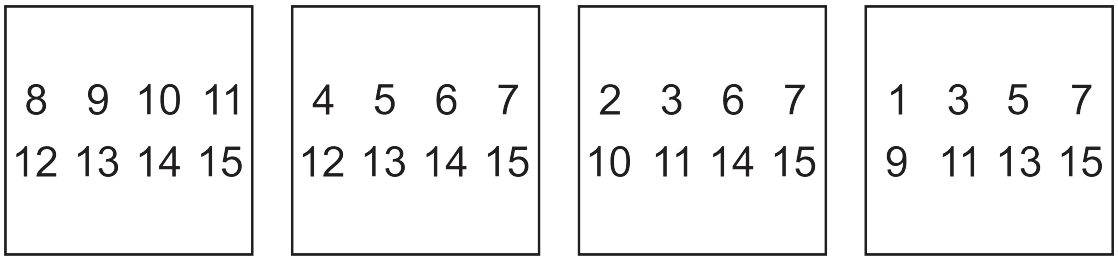


Figura 1.2: Truque com Cartões

É evidente que o truque funciona para qualquer potência de 2, note porém que para cada expoente de 2 é necessário uma cartela. Considerando que a última cartela terá como primeiro valor a potência 2^n, então teremos $n+1$ cartelas, iniciadas com 2^0, 2^1, 2^2, ... , 2^{n-1}, 2^n e, para escrever o máximo de valores sem atingir a potência seguinte, podemos escrever até o número $2^{n+1}-1$.

Um exemplo desse jogo expandido para números até 63 é:

32	33	34	35	36	37
38	39	40	41	42	43
44	45	46	47	48	49
50	51	52	53	54	55
56	57	58	59	60	61
62	63				

16	17	18	19	20	21
22	23	24	25	26	27
28	29	30	31	48	49
50	51	52	53	54	55
56	57	58	59	60	61
62	63				

8	9	10	11	12	13
14	15	24	25	26	27
28	29	30	31	40	41
42	43	44	45	46	47
56	57	58	59	60	61
62	63				

4	5	6	7	12	13
14	15	20	21	22	23
28	29	30	31	36	37
38	39	44	45	46	47
52	53	54	55	60	61
62	63				

2	3	6	7	10	11
14	15	18	19	22	23
26	27	30	31	34	35
38	39	42	43	46	47
50	51	54	55	58	59
62	63				

1	3	5	7	9	11
13	15	17	19	21	23
25	27	29	31	33	35
37	39	41	43	45	47
49	51	53	55	57	59
61	63				

Figura 1.3: Cartões com números de 1 a 63

A forma de confeccionar os cartões é a mesma e o truque funciona igualmente.

Suponha um voluntário escolhendo o Primeiro, Terceiro, Quinto e Sexto cartão (da maior para a menor potência).

Certamente o valor pensado por ele será o 43, pois temos: $32 + 8 + 2 + 1 = 43$

Veja na tabela a seguir os números de 0 a 63 escritos em binários.

	Binários							Binários						
Decimais	2^5	2^4	2^3	2^2	2^1	2^0	Decimais	2^5	2^4	2^3	2^2	2^1	2^0
0						0	32	1	0	0	0	0	0
1						1	33	1	0	0	0	0	1
2					1	0	34	1	0	0	0	1	0
3					1	1	35	1	0	0	0	1	1
4				1	0	0	36	1	0	0	1	0	0
5				1	0	1	37	1	0	0	1	0	1
6				1	1	0	38	1	0	0	1	1	0
7				1	1	1	39	1	0	0	1	1	1
8			1	0	0	0	40	1	0	1	0	0	0
9			1	0	0	1	41	1	0	1	0	0	1
10			1	0	1	0	42	1	0	1	0	1	0
11			1	0	1	1	43	1	0	1	0	1	1
12			1	1	0	0	44	1	0	1	1	0	0
13			1	1	0	1	45	1	0	1	1	0	1
14			1	1	1	0	46	1	0	1	1	1	0
15			1	1	1	1	47	1	0	1	1	1	1
16		1	0	0	0	0	48	1	1	0	0	0	0
17		1	0	0	0	1	49	1	1	0	0	0	1
18		1	0	0	1	0	50	1	1	0	0	1	0
19		1	0	0	1	1	51	1	1	0	0	1	1
20		1	0	1	0	0	52	1	1	0	1	0	0
21		1	0	1	0	1	53	1	1	0	1	0	1
22		1	0	1	1	0	54	1	1	0	1	1	0
23		1	0	1	1	1	55	1	1	0	1	1	1
24		1	1	0	0	0	56	1	1	1	0	0	0
25		1	1	0	0	1	57	1	1	1	0	0	1
26		1	1	0	1	0	58	1	1	1	0	1	0
27		1	1	0	1	1	59	1	1	1	0	1	1
28		1	1	1	0	0	60	1	1	1	1	0	0
29		1	1	1	0	1	61	1	1	1	1	0	1
30		1	1	1	1	0	62	1	1	1	1	1	0
31		1	1	1	1	1	63	1	1	1	1	1	1

Tabela 1.1: Números binários de 0 a 63

Esta tabela pode ser útil ao "Mago" caso queira mostrar a sua plateia ao final de tudo, como o truque funcionou.

1.2 Códigos com Dígito de Checagem

"Generalizemos para simplificar, ou para compreender melhor!"
Jacques Hadamard

Na introdução fizemos uma analogia entre o nosso idioma e códigos. Pessoas muitas vezes podem detectar erros quando apresentados com novas informações (redundâncias). No entanto, os computadores não possuem essa intuição e exigem métodos de verificação de erros ao transferir informações. O mais simples desses métodos é chamado de "paridade", e consiste em adicionar redundâncias aos códigos a serem transmitidos.

Definição 1.5. Uma *Matriz de Código Binário* é uma tabela retangular (em formato de linha ou coluna) de números chamados componentes da matriz, onde cada componente é o número 0 (zero) ou 1 (um).

São exemplos de Matrizes de Código Binário:

$$A = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad e \quad D = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}$$

Suponha que tenhamos codificado uma mensagem como um conjunto de matrizes de códigos binários (baseado em [14]). Queremos enviar essas matrizes através de um *canal*, mas infelizmente, o canal pode ser "ruidoso". Como consequência, podem ocorrer erros: alguns 0s podem ser trocados por 1s, e vice-versa. Como podemos precaver desse problema?

Exemplo 1.6. Desejamos codificar e transmitir uma mensagem que consiste em uma das palavras: *cima*, *baixo*, *esquerda* ou *direita*. Decidimos usar os quatro matrizes como nosso código binário, observe:

Mensagem	Cima	Baixo	Esquerda	Direita
Código	[0 0]	[0 1]	[1 0]	[1 1]

Tabela 1.2: Mensagem Codificada

Se o receptor também tiver essa tabela e a mensagem codificada for transmitida sem erros, decodificar será trivial. No entanto, vamos admitir que tenha ocorrido um erro simples. *Erro simples* é uma alteração em exatamente uma coordenada da matriz de código.

Suponha o envio da mensagem “Baixo”, codificada como $[\ 0 \quad 1\ ]$, com um engano na transmissão da primeira coordenada em que o 0 foi trocado por 1. O receptor verá então a mensagem $[\ 1 \quad 1\ ]$ e a decodificará como “Direita”. Veja no diagrama:

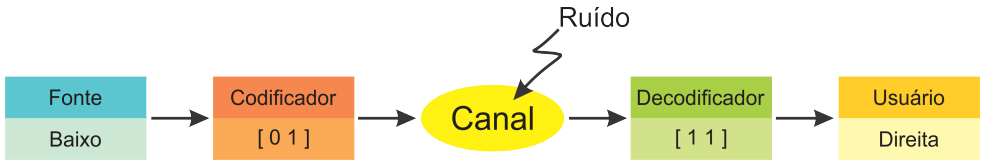


Figura 1.4: Esquema de codificação e decodificação

Agora, mesmo que o receptor percebesse (de alguma maneira) que ocorreu um erro simples, ele não saberia se a matriz correta era $[\ 0 \quad 1\ ]$ ou $[\ 1 \quad 0\ ]$.

Se recebido $[\ 0 \quad 0\ ]$ a mensagem enviada poderia ter sido $[\ 0 \quad 0\ ]$ (sem ruído no canal), $[\ 0 \quad 1\ ]$ ou $[\ 1 \quad 0\ ]$, com ruído nos dois últimos casos.

Suponha o envio da mensagem com um código binário como mostra a tabela a seguir:

Mensagem	Cima	Baixo	Esquerda	Direita
Código	$[\ 0 \quad 0 \quad 0\ ]$	$[\ 0 \quad 1 \quad 1\ ]$	$[\ 1 \quad 0 \quad 1\ ]$	$[\ 1 \quad 1 \quad 0\ ]$

Tabela 1.3: Mensagem Codificada

Esse código poderá detectar qualquer erro simples. Por exemplo, se “Baixo” for enviado como $[\ 0 \quad 1 \quad 1\ ]$ e ocorrer um erro em alguma coordenada, o receptor lerá $[\ 1 \quad 1 \quad 1\ ]$, $[\ 0 \quad 0 \quad 1\ ]$ ou $[\ 0 \quad 1 \quad 0\ ]$, e nenhum deles é uma matriz de código. Assim, ele saberá que ocorreu um erro (mas não onde), e poderá pedir que a mensagem codificada seja retransmitida (o que pode gerar novos custos de envio).

Por que o receptor não saberá onde está o erro?

Observe que a matriz $[\ 1 \quad 1 \quad 1\ ]$ pode ser uma variação de “Esquerda” $[\ 1 \quad 0 \quad 1\ ]$ ou “Direita” $[\ 1 \quad 1 \quad 0\ ]$, e que a matriz $[\ 0 \quad 0 \quad 1\ ]$ pode ser uma variação de “Cima” $[\ 0 \quad 0 \quad 0\ ]$, “Esquerda” $[\ 1 \quad 0 \quad 1\ ]$, e que a matriz $[\ 0 \quad 1 \quad 0\ ]$ pode ser uma variação de “Cima” $[\ 0 \quad 0 \quad 0\ ]$ ou “Direita” $[\ 1 \quad 1 \quad 0\ ]$.

O código da Tabela 1.3 é um exemplo de *código detector de erros*. Até os anos 40, isso era o melhor que podia ser obtido. A criação dos computadores digitais conduziu ao desenvolvimento de códigos que podiam *corrigir* tão bem quanto detectar erros.

A própria mensagem a ser transmitida pode ser formada por matrizes de código binário. Neste caso, um código detector de erros simples, mas muito usado é o *código de checagem de paridade*, que é obtido anexando-se uma nova componente - chamada *código de checagem* - a

cada matriz, de maneira que a paridade (o número total de 1s) seja um número par, foi dessa forma que construímos o código da Tabela 1.3.

Para formalizar este conceito suponha que a mensagem seja a matriz de código binário

$$\mathbf{b} = [\ b_1 \quad b_2 \quad \cdots \quad b_n\ ]\ ,\ b_i \in \{0,1\}.$$

Então, a matriz código de checagem de paridade é $\mathbf{v} = [\ b_1 \quad b_2 \quad \cdots \quad b_n \quad d\ ]$, onde o código de checagem d é escolhido de maneira que

$$b_1 + b_2 + \cdots + b_n + d \qquad \text{seja um número par}$$

Exemplo 1.7. Se a mensagem a ser enviada for a matriz de código binário $[\ 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1\ ]$, que possui um número ímpar de 1s, o dígito de checagem será 1 e a matriz de código será

$$[\ 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1\ ]$$

Note que qualquer erro simples será detectado, pois ele mudará a paridade da matriz de código de par para ímpar. Supondo que ocorra um erro na terceira coordenada, a matriz de código será recebida como $[\ 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1\ ]$, para paridade é ímpar, pois tem cinco 1s. Veja o diagrama.

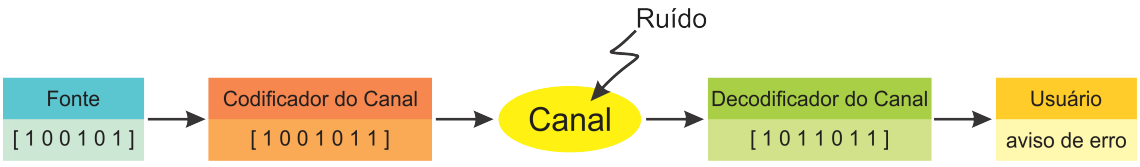


Figura 1.5: Paridade

1.2.1 Um Truque com Paridade

Como a verificação de paridade é simples de ser feita, podemos aplicá-la a séries iniciais através de atividades educacionais direcionadas. Vejamos um truque de adivinhação, que se encontra em [1], que aplica de forma recreativa o conceito de par e ímpar.

Esse truque pode ser aplicado de forma simples em sala de aula. Esta atividade utiliza um truque de mágica para mostrar como detectar e corrigir erros quando dados foram corrompidos.

O truque requer uma pilha de cartas idênticas de dois lados. As cartas podem ser vermelhas

de um lado e brancas no outro, por exemplo. Uma maneira fácil de fazê-las é cortar uma grande folha de papel-cartão que é colorida em um lado apenas.

Figura 1.6: Papel-cartão

Um pacote de cartas de baralho também é adequado.

Figura 1.7: Baralho

Etapas do truque

1. Peça a uma ou duas crianças que coloquem as cartas, em forma retangular. As crianças podem decidir aleatoriamente qual a posição para cada carta. Veja um exemplo de retângulo aleatório de 5x5 cartas (Figura 1.8).

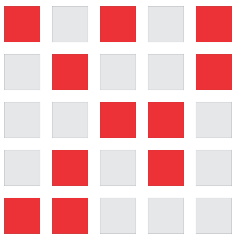


Figura 1.8: Retângulo 5x5 - Inicial

2. Casualmente, adicione outra linha e coluna ao retângulo "apenas para torná-lo um pouco mais difícil" (Figura 1.9). Claro, essas cartas são a chave para o truque. A estratégia é escolher as cartas extras para garantir que haja um número par de cartas coloridas em cada linha e coluna.

3. Selecione uma criança e, enquanto você cobrir seus olhos e/ou desviar o olhar, peça para a criança virar uma carta - apenas uma carta. Na Figura 1.10 temos um exemplo, a quarta carta na quarta linha foi invertida. Em seguida você descobre seus olhos, estuda as cartas e identifica qual delas foi virada.

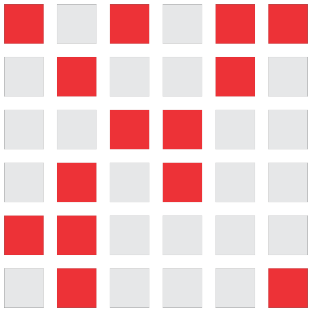


Figura 1.9: Retângulo 6x6 - Com os cartas extras

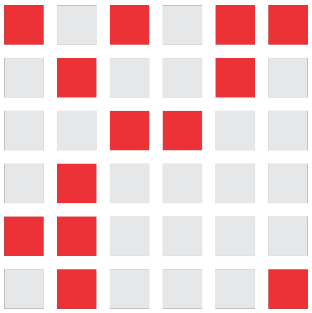


Figura 1.10: Retângulo 6x6 - Com uma carta virada

Como adivinhar?

Simples! Devido à forma como as cartas foram colocadas, a linha e a coluna que contém a carta alterada terão agora um número ímpar de cartas coloridas, o que identifica rapidamente a carta virada.

Na conclusão, diga às crianças que as cartas de paridade são usadas para mostrar a ocorrência de um erro.

No capítulo “Modelos de Atividades Propostas” temos essa atividade descrita com mais detalhes.

Capítulo 2

Códigos de Hamming

"— Quem te recitou todas essas coisas tão difíceis?
— Li eu num livro - respondeu Alice."
Lewis Carroll, Alice do Outro Lado do Espelho

Os códigos corretores de erros tem a função de detectar e corrigir erros que surjam na transmissão ou armazenamento de dados [8].

O advento dos computadores digitais do século XX levou à necessidade de transmitir rapidamente e com precisão quantidades maciças de dados. Em um meio eletrônico, o alfabeto é composto por dois símbolos 0 e 1. Ele é chamado o alfabeto binário e representado por $\mathbb{Z}_2$, isto é, $\mathbb{Z}_2 = \{0, 1\}$. Neste capítulo vamos abordar o conhecimento sobre códigos e como a Álgebra pode ajudar detectar e corrigir erros que surjam em sua transmissão.

2.1 Preliminares

2.1.1 O conjunto $\mathbb{Z}_2$

Já que computadores descrevem dados em termos de 0s e 1s (falso/verdadeiro ou não/sim), começamos considerando códigos *binários*, que consistem em Matrizes Linha ou Matrizes Coluna, cujas coordenadas são iguais a 0 ou 1. Nesse contexto, as regras usuais da aritmética devem ser modificadas, pois o resultado de cada cálculo que envolvem escalares deve ser 0 ou 1.

Veja a seguir as regras modificadas para adição e multiplicação.

$+$	0	1
0	0	1
1	1	0

Tabela 2.1: Soma em $\mathbb{Z}_2$

$\times$	0	1
0	0	0
1	0	1

Tabela 2.2: Multiplicação em $\mathbb{Z}_2$

A única curiosidade aqui é o fato de que $1+1=0$. Isto não é tão estranho quanto parece: se substituirmos 0 pela palavra "par" e 1 pela palavra "ímpar", essas tabelas simplesmente resumem as *regras de paridade* para a adição e multiplicação de inteiros pares e ímpares. Por exemplo, $1+1=0$ expressa a propriedade de que a soma de dois inteiros ímpares é um inteiro par.

O conjunto formado somente pelos números 0 e 1, onde 0 representa todos os números que divididos por 2 deixam resto 0 e 1 representa todos os números que divididos por 2 deixam resto 1 é denotado por $\mathbb{Z}_2$ e é chamado de conjunto dos *inteiros módulo 2*.

Exemplo 2.1. Em $\mathbb{Z}_2$, $1+1+0+1=1$ e $1+1+1+1=0$. (novamente temos as regras de paridade: a soma de três ímpares é ímpar; a soma de quatro ímpares é par.)

2.1.2 Matrizes de Código em $\mathbb{Z}_2$

Com $\mathbb{Z}_2$ como conjunto numérico, podemos estender para Matrizes as regras de paridade.

São exemplos de Matrizes de Código Binário:

$$A=\begin{bmatrix}1&0&1\end{bmatrix},\quad B=\begin{bmatrix}1\\1\\0\\1\end{bmatrix},\quad C=\begin{bmatrix}1\\0\\1\\1\\1\\0\\1\end{bmatrix}\quad e\quad D=\begin{bmatrix}0&0&1&1\end{bmatrix}$$

Definição 2.2. Uma *Matriz Binária* é uma tabela retangular (com m linhas e n colunas) de números chamados componentes da matriz, onde cada componente é o número 0 (zero) ou 1 (um).

Estes são exemplos de matrizes binárias.

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \; e \; \begin{bmatrix} 1 \end{bmatrix}$$

São exemplos de Matrizes de Código as seguintes matrizes:

$$A = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad e \quad D = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}$$

A Adição de Matrizes permanece com suas propriedades usuais:

São as Propriedades da Adição de Matrizes:

$$\begin{array}{lll} (i) & A + B = B + A & \text{Propriedade Comutativa} \\ (ii) & A + (B + C) = (A + B) + C & \text{Propriedade Associativa} \end{array}$$

Exemplo de soma de matrizes em $\mathbb{Z}_2$

Exemplo 2.3. Se $A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ e $B = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$, então

$$A + B = \begin{bmatrix} 1+0 & 1+1 & 1+1 & 0+1 \\ 0+1 & 0+0 & 1+1 & 1+0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Definição 2.4. Uma matriz quadrada cujos elementos fora da diagonal são todos zero e que todos os elementos da diagonal são iguais a 1 é chamada de *matriz identidade* e I_k é a matriz identidade de ordem k.

Da mesma, forma em $\mathbb{Z}_2$ as propriedades usuais da Multiplicação de Matrizes permanecem. Seguem as Propriedades da multiplicação de Matrizes:

Sejam A, B e C matrizes (cujas ordens possibilitem que as operações indicadas sejam reali-

zadas). Então:

$$
\begin{array}{lcl}
(i) & A(BC) = (AB)C & \text{Associativa} \\
(ii) & A(B+C) = AB + AC & \text{Distributiva à esquerda} \\
(iii) & (A+B)C = AC + BC & \text{Distributiva à direita} \\
(iv) & AI = IA = A & \text{Identidade da multiplicação}
\end{array}
$$

A Demonstração das Propriedades de Soma e Multiplicação de Matrizes pode ser encontrada em [14].

Segue um exemplo de produto de Matrizes em $\mathbb{Z}_2$.

Exemplo 2.5. Calcule, em $\mathbb{Z}_2$, $A \cdot B$, dados

$$
A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}
$$

Solução:

$$
A \cdot B = \begin{bmatrix} (1+1+0+0) & (1+0+0+0) & (0+1+0+0) & (1+1+0+0) \\ (0+1+0+1) & (0+0+1+0) & (0+1+1+0) & (0+1+1+1) \end{bmatrix}
$$

Assim, a matriz produto é:

$$
A \cdot B = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}
$$

Primeiramente usaremos essas matrizes para projetar códigos que detectam erros que podem ocorrer na transmissão de uma mensagem. Esta mensagem pode ser, por exemplo: palavras, números ou símbolos. Posteriormente, construiremos códigos que não somente detectam, mas corrigem erros. Começaremos transformando cada "palavra" da mensagem em uma matriz de código binário, ou seja, codificando a "palavra".

Definição 2.6. Um *código binário* é um conjunto de matrizes de código binário (de mesmo comprimento) chamadas *matrizes de código*.

As Matrizes Linha em $\mathbb{Z}_2^n$ são chamadas *Matrizes Binárias de Comprimento n.*

Exemplo 2.7. As Matrizes em $\mathbb{Z}_2^3$ são todas as matrizes de código binário de comprimento 3. São elas:

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

Como veremos, é importante que um código tenha outras propriedades, como a habilidade de detectar quando ocorre um erro na transmissão de uma matriz de código, e, se possível, sugerir como corrigir o erro.

2.2 O Peso de Hamming e a Distância de Hamming

Em [16] vemos que para iniciarmos a construção de um código corretor de erros devemos tomar um conjunto finito A, chamado de *alfabeto*. Aqui teremos nosso "alfabeto" em $\mathbb{Z}_2$. Quando transmitimos símbolos de uma mensagem um a um, não há como detectar um erro, por mais simples que ele seja. É por isso que, para transmiti-la, vamos tentar dividir a mensagem em blocos de símbolos de comprimento fixo n.

O conjunto de todas as matrizes de código binário de dimensão n, será denotado por $\mathbb{Z}_2^n$. Um subconjunto de $\mathcal{C}$ de $\mathbb{Z}_2^n$ é chamado de código, assim cada elemento do *código corretor de erros* é chamado palavra-código assim cada mensagem pode ser representada por uma matriz $[\ a_1 \quad a_2 \quad \cdots \quad a_n\ ] \in \mathbb{Z}_2^n$.

Os erros durante a transmissão também podem ser modelados algebricamente. Suponha que uma mensagem $\mathbf{u} = [\ u_1 \quad u_2 \quad \cdots \quad u_n\ ] \in \mathbb{Z}_2^n$ foi transmitida e $\mathbf{v} = [\ v_1 \quad v_2 \quad \cdots \quad v_n\ ] \in \mathbb{Z}_2^n$ foi recebida com um erro na posição i, então

$$\mathbf{v} = \mathbf{u} + e_i$$

Onde $e_i = [\ 0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0\ ]$ com um 1 na i-ésima posição e 0 nas outras posições.

Exemplo 2.8. Supondo que ao enviar $u = [\ 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1\ ]$ haja um erro na quinta posição, teríamos $e_5 = [\ 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0\ ]$ logo:

$$\mathbf{v} = \mathbf{u} + e_5 = [\ 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1\ ] + [\ 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0\ ] = [\ 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1\ ]$$

Caso aconteça das posições $i_1, i_2, \cdots, i_k$ serem danificadas, então

$$\mathbf{v} = \mathbf{u} + \mathbf{e}$$

Onde $P(\mathbf{e}) = e_{i_1} + e_{i_2} + \cdots + e_{i_k}$ é uma matriz com k uns e $n-k$ zeros. Neste caso ela é chamada de *matriz de erro*.

Definição 2.9. O *peso* (Hamming) de uma matriz $\mathbf{u} \in \mathbb{Z}_2^n$ é o número de coordenadas não nulas em $\mathbf{u}$ e o denotamos por $P(\mathbf{u})$

Se uma mensagem $\mathbf{u} = [\ u_1 \quad u_2 \quad \cdots \quad u_n\ ] \in \mathbb{Z}_2^n$ foi transmitida e $\mathbf{v} = [\ v_1 \quad v_2 \quad \cdots \quad v_n\ ] \in \mathbb{Z}_2^n$ foi recebida com k erros durante a transmissão, então $\mathbf{v} = \mathbf{u} + e$ com $P(\mathbf{e}) = k$.

Exemplo 2.10. Se $\mathbf{u} = [\ 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1\ ]$ e $\mathbf{v} = [\ 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1\ ]$,

então $e = [\ 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0\ ]$, com $P(\mathbf{e}) = 3$.

Observe que se pretendêssemos escrever a palavra "vaca", mas acidentalmente fosse escrita como "maca", ou como "faca", ou ainda "vaia", o erro seria em apenas uma coordenada e, mesmo assim, não poderia ser detectado por um receptor. A proximidade entre essas palavras dificulta que se detecte o erro.

A fim de verificarmos com precisão a proximidade entre a *mensagem enviada* e a *mensagem recebida com erro* definimos um outro conceito importante: a *distância de Hamming*.

Definição 2.11. Dados dois elementos $\mathbf{u} = [\ u_1 \quad u_2 \quad \cdots \quad u_n\ ]$, $\mathbf{v} = [\ v_1 \quad v_2 \quad \cdots \quad v_n\ ] \in \mathbb{Z}_2^n$, a *distância de Hamming* entre $\mathbf{u}$ e $\mathbf{v}$ é o número de coordenadas em que essas duas matrizes diferem, que denotamos por $d(\mathbf{u},\mathbf{v}) = |i \in \{1, 2, 3, \cdots, \text{n}\}\ ;\ u_i \neq v_i|$.

Para não sobrecarregar a notação, ao calcular a distância de Hamming, não usaremos os colchetes e os espaços entre coordenadas das matrizes.

Exemplo 2.12. Sejam as matrizes em $\mathbb{Z}_2^3$: $\mathbf{a} = [\ 0 \quad 0 \quad 0\ ]$, $\mathbf{b} = [\ 0 \quad 0 \quad 1\ ]$, $\mathbf{c} = [\ 0 \quad 1 \quad 0\ ]$, $\mathbf{d} = [\ 1 \quad 0 \quad 1\ ]$, $\mathbf{e} = [\ 1 \quad 1 \quad 1\ ]$

Calculando a distância de Hamming temos:

$$d(\mathbf{a},\mathbf{b}) = d(000, 001) = 1;$$

$$d(\mathbf{c},\mathbf{d}) = d(010, 101) = 3;$$

$$d(\mathbf{e},\mathbf{a}) = d(111, 001) = 2.$$

Supondo que uma mensagem $\mathbf{u} = [\ u_1 \quad u_2 \quad \cdots \quad u_n\ ] \in \mathbb{Z}_2^n$ foi transmitida e $\mathbf{v} = [\ v_1 \quad v_2 \quad \cdots \quad v_n\ ] \in \mathbb{Z}_2^n$ foi recebida. Então o fato de que k erros ocorrem durante a transmissão é equivalente a dizer que $d(\mathbf{u},\mathbf{v}) = k$.

Proposição 2.13. As propriedades usuais de distância são válidas para a *distância de Hamming*.

Dados $\mathbf{u}$, $\mathbf{v}$ e $\mathbf{t} \in \mathbb{Z}_2^n$, temos:

(i) *Positividade* : $d(\mathbf{u},\mathbf{v}) \geq 0$ e $d(\mathbf{u},\mathbf{v}) = 0$ se, e somente se $\mathbf{u} = \mathbf{v}$;

(ii) *Simetria* : $d(\mathbf{u},\mathbf{v}) = d(\mathbf{v},\mathbf{u})$;

(iii) *Desigualdade Triangular* : $d(\mathbf{u},\mathbf{v}) \leq d(\mathbf{u},\mathbf{t}) + d(\mathbf{t},\mathbf{v})$.

Demonstração. A prova das duas primeiras é direta. Vamos à prova da terceira: Suponha que $u_i \neq v_i$ e a posição i contribui 1 para $d(\mathbf{u},\mathbf{v})$. Então, ou $u_i = t_i$ e $t_i \neq v_i$ ou $u_i \neq t_i$ e $t_i = v_i$. Assim, a i-ésima posição também contribuirá 1 à soma $d(\mathbf{u},\mathbf{t}) + d(\mathbf{t},\mathbf{v})$. Suponha agora que $u_i = v_i$ e a posição i contribui 0 para $d(\mathbf{u},\mathbf{v})$. Então $u_i = v_i = t_i$ e a i-ésima posição contribuem também 0 à soma $d(\mathbf{u},\mathbf{t}) + d(\mathbf{t},\mathbf{v})$ ou $u_i \neq v_i \neq t_i$ e a i-ésima posição contribui 2 à soma $d(\mathbf{u},\mathbf{t}) + d(\mathbf{t},\mathbf{v})$. Assim, $d(\mathbf{u},\mathbf{v}) \leq d(\mathbf{u},\mathbf{t}) + d(\mathbf{t},\mathbf{v})$. □

Uma vez definida a distância, no conjunto das matrizes de código binário de comprimento n, podemos definir conceitos geométricos neste conjunto.

Os conjuntos a seguir desempenham um papel fundamental na teoria de códigos.

Para qualquer $\mathbf{u} \in \mathbb{Z}_2^n$ e um número real $k > 0$, definimos $B(\mathbf{u}, k) = \{\mathbf{v} \in \mathbb{Z}_2^n : d(\mathbf{v},\mathbf{u}) \leq k\}$ e chamamos de *bola* de raio k com centro em $\mathbf{u}$.

Exemplo 2.14. Seja $\mathbf{u} = [\ 1 \quad 1 \quad 1 \quad 1\ ] \in \mathbb{Z}_2^4$,

Então $B(\mathbf{u}, 2) = \{\mathbf{u}, [\ 0 \quad 1 \quad 1 \quad 1\ ], [\ 1 \quad 0 \quad 1 \quad 1\ ], [\ 1 \quad 1 \quad 0 \quad 1\ ], [\ 1 \quad 1 \quad 1 \quad 0\ ],$

$[\ 0 \quad 0 \quad 1 \quad 1\ ], [\ 0 \quad 1 \quad 0 \quad 1\ ], [\ 0 \quad 1 \quad 1 \quad 0\ ], [\ 1 \quad 0 \quad 0 \quad 1\ ], [\ 1 \quad 0 \quad 1 \quad 0\ ], [\ 1 \quad 1 \quad 0 \quad 0\ ]\}$

Teorema 2.15. O *número de elementos* da bola de raio k com centro $\mathbf{u}$ é dado por:

$$|B(\mathbf{u}, k)| = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} \cdots \binom{n}{k}.$$

Demonstração. Seja $\mathbf{v} \in B(\mathbf{u}, k)$ e consideremos a "matriz de erro" e tal que $\mathbf{v} = \mathbf{u} + e$. Então $\mathbf{v} \in B(\mathbf{u}, k)$ se, e somente se, $P(e) \leq k$.

Basta provar que, para cada $i = (1, 2, \cdots, k)$, há exatamente $\binom{n}{i}$ matrizes $e \in \mathbb{Z}_2^n$ tal que $P(e) = i$.

Como e é matriz de erro, temos:

Há uma matriz e, tal que $\mathbf{v} = \mathbf{u} + e$, com $e = [\ 0 \quad 0 \quad 0 \quad \cdots \quad 0\ ]$, com n coordenadas iguais a zero, chamada matriz zero, e $P(e) = 0$.

Há n matrizes e distintas, tal que $\mathbf{v} = \mathbf{u} + e$, com $e = [\ 0 \quad \cdots \quad 1 \quad \cdots \quad 0\ ]$, ou seja n posições para um único número 1 ocupar, e $P(e) = 1$.

Agora e se ouvessem 2 erros na matriz $\mathbf{v}$? Teríamos o número 1 em duas posições de e dentre as n possíveis, sendo $P(e) = 2$. Isso quer dizer que temos $\binom{n}{2}$ matrizes e distintas.

E se ouvessem 3 erros na matriz $\mathbf{v}$, seriam $\binom{n}{3}$ matrizes e distintas, cada uma delas com peso 3.

Na verdade, devemos escolher i posições de n na matriz zero e alterar as coordenadas para 1s. Assim, cada matriz e com $P(e) = i$ corresponde a um subconjunto de i-elementos de $\{1, 2, \cdots, n\}$. Sabemos que existem

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

subconjuntos. Agora é claro que a fórmula

$$|B(\mathbf{u}, k)| = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} \cdots \binom{n}{k}$$

conta todas as "matriz de erro" de peso no máximo k, e portanto todas as matrizes $\mathbf{v}$ que estão à distância de Hamming k ou menos de $\mathbf{u}$. □

Eis um exemplo numérico:

Exemplo 2.16. A quantidade de elementos da bola de raio 2 com centro $\mathbf{u} = [\ 1 \quad 1 \quad 1 \quad 1\ ] \in \mathbb{Z}_2^4$, é igual a:

$$|B(\mathbf{u}, 2)| = \binom{4}{0} + \binom{4}{1} + \binom{4}{2} = 1 + 4 + \frac{4(4-1)}{2} = 1 + 4 + 6 = 11.$$

Ou seja, há 11 elementos que pertencem a bola de raio 2 e centro em $\mathbf{u} = [\ 1 \quad 1 \quad 1 \quad 1\ ] \in \mathbb{Z}_2^4$.

2.3 Distância de Hamming Mínima

Definição 2.17. Seja $\mathcal{C} \subseteq \mathbb{Z}_2^n$ um código. A *distância de Hamming mínima* de $\mathcal{C}$ é:

$$d = min\{d(\mathbf{u},\mathbf{v}) : \mathbf{u},\mathbf{v} \in \mathcal{C} \text{ e } \mathbf{u} \neq \mathbf{v}\}$$

Ou seja, a *distância de Hamming mínima* em um código é a menor distância entre duas palavras de código distintas, sobre todos os pares de palavras de código.

Exemplo 2.18. Considere o código $\mathcal{C} = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ onde:

$$\mathbf{a} = [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ ],\ \mathbf{b} = [\ 1\ \ 0\ \ 1\ \ 1\ \ 0\ ],$$

$$\mathbf{c} = [\ 0\ \ 1\ \ 0\ \ 1\ \ 1\ ],\ \mathbf{d} = [\ 1\ \ 1\ \ 1\ \ 0\ \ 1\ ].$$

Calculando a distância de Hamming entre cada par de palavras de código, temos:

$$d(\mathbf{a},\ \mathbf{b}) = d(00000, 10110) = 3$$

$$d(\mathbf{a},\ \mathbf{c}) = d(00000, 01011) = 3$$

$$d(\mathbf{a},\ \mathbf{d}) = d(00000, 11101) = 4$$

$$d(\mathbf{b},\ \mathbf{c}) = d(10110, 01011) = 4$$

$$d(\mathbf{b},\ \mathbf{d}) = d(10110, 11101) = 3$$

$$d(\mathbf{c},\ \mathbf{d}) = d(01011, 11101) = 3$$

Portanto este código tem distância de Hamming mínima $d = 3$.

Exemplo 2.19. Considere o código $\mathcal{C} = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}\}$ onde:

$$\mathbf{a} = [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ ],\ \mathbf{b} = [\ 0\ \ 1\ \ 1\ \ 1\ \ 0\ ],\ \mathbf{c} = [\ 1\ \ 0\ \ 1\ \ 0\ \ 1\ ],$$

$$\mathbf{d} = [\ 1\ \ 1\ \ 0\ \ 1\ \ 1\ ],\ \mathbf{e} = [\ 1\ \ 1\ \ 1\ \ 1\ \ 1\ ]$$

Calculando a distância de Hamming entre cada par de palavras de código, temos:

$$d(\mathbf{a},\ \mathbf{b}) = d(00000, 01110) = 3$$

$$d(\mathbf{a},\ \mathbf{c}) = d(00000, 10101) = 3$$

$$d(\mathbf{a},\ \mathbf{d}) = d(00000, 11011) = 4$$

$$d(\mathbf{a},\ \mathbf{e}) = d(00000, 11111) = 5$$

$$d(\mathbf{b},\ \mathbf{c}) = d(01110, 10101) = 4$$

$$d(\mathbf{b},\ \mathbf{d}) = d(01110, 11011) = 3$$

$$d(\mathbf{b},\ \mathbf{e}) = d(01110, 11111) = 2$$

$$d(\mathbf{c},\ \mathbf{d}) = d(10101, 11011) = 3$$

$$d(\mathbf{c},\ \mathbf{e}) = d(10101, 11111) = 2$$

$$d(\mathbf{d},\ \mathbf{e}) = d(11011, 11111) = 1$$

Logo este código tem distância de Hamming mínima $d = 1$.

Como visto a partir destes dois exemplos, se quisermos calcular a distância de Hamming mínima entre as palavras de um código, é necessário verificar $\binom{M}{2}$ pares de palavras de código para encontrar o par com a menor distância, onde M é o número de palavras do código $\mathcal{C}$. Note que esta tarefa pode ser nada fácil e ainda haver muitos pares que dão este mínimo.

Dado um código $\mathcal{C}$ com distância mínima d, define-se

$$k = \left\lfloor \frac{d-1}{2} \right\rfloor,$$

onde notação $\lfloor t \rfloor$ designa o maior inteiro que é menor ou igual a t. Por exemplo, $\lfloor 17,54 \rfloor = 17$, $\lfloor \pi \rfloor = 3$.

Lema 2.20. Seja $\mathcal{C}$ um código com distância mínima d. Se $\mathbf{u}$ e $\mathbf{v}$ são palavras distintas de $\mathcal{C}$, então

$$B(\mathbf{u}, k) \cap B(\mathbf{v}, k) = \emptyset$$

Demonstração. De fato, $\mathbf{x}$ pertencesse a $B(\mathbf{u}, k) \cap B(\mathbf{v}, k)$, teríamos

$$d(\mathbf{x}, \mathbf{u}) \leq k \quad \text{e} \quad d(\mathbf{x}, \mathbf{v}) \leq k$$

e portanto, pela simetria es pela desigualdade triangular,

$$d(\mathbf{u}, \mathbf{v}) \leq d(\mathbf{u}, \mathbf{x}) + d(\mathbf{x}, \mathbf{v}) \leq k + k = 2k = 2 \left\lfloor \frac{d-1}{2} \right\rfloor \leq d - 1$$

Logo encontramos uma contradição, pois $d(\mathbf{u}, \mathbf{v}) \geq d$. □

O processo de conversão de uma mensagem em matrizes de código é chamado *codificação*, e o processo inverso é chamado de *decodificação*.

Um código é chamado de *corretor de erros* se um decodificador é capaz de corrigir qualquer padrão de k ou menos erros introduzidos pelo canal. Neste caso, ao receber um palavra de código qualquer, que tenha sido alterada em k ou menos posições de coordenadas, o decodificador consegue recuperar a mensagem verdadeira.

Exemplo 2.21. Vamos supor que um código $\mathcal{C}$ que contenha as seguintes palavras.

$$[\,0\;0\;0\;1\;1\;1\;1\,]\;,\;[\,0\;0\;1\;0\;0\;1\;1\,]\;,\;[\,0\;0\;1\;1\;1\;0\;0\,]\;,$$
$$[\,0\;1\;0\;0\;1\;0\;1\,]\;,\;[\,0\;1\;0\;1\;0\;1\;0\,]\;,\;[\,0\;1\;1\;0\;1\;1\;0\,]\;,$$
$$[\,0\;1\;1\;1\;0\;0\;1\,]\;,\;[\,1\;0\;0\;0\;1\;1\;0\,]\;,\;[\,1\;0\;0\;1\;0\;0\;1\,]\;,$$
$$[\,1\;0\;1\;0\;1\;0\;1\,]\;,\;[\,1\;0\;1\;1\;0\;1\;0\,]\;,\;[\,1\;1\;0\;0\;0\;1\;1\,]\;,$$
$$[\,1\;1\;0\;1\;1\;0\;0\,]\;,\;[\,1\;1\;1\;0\;0\;0\;0\,]\;,\;[\,1\;1\;1\;1\;1\;1\;1\,]$$

O código $\mathcal{C}$ tem distância de Hamming mínima $d = 3$, logo $k = 1$.

Agora supondo que ao enviar uma das palavras-código, tenha sido recebida a palavra

$$[\,1\;1\;1\;1\;0\;1\;0\,]$$

Calculando a distância das palavras-código à palavra recebida temos:

$$d(0001111, 1111010) = 5$$

$$d(0010011, 1111010) = 4$$

$$d(0011100, 1111010) = 4$$

$$d(0100101, 1111010) = 5$$

$$d(0101010, 1111010) = 2$$

$$d(0110110, 1111010) = 3$$

$$d(0111001, 1111010) = 2$$

$$d(1000110, 1111010) = 4$$

$$d(1001001, 1111010) = 3$$

$$d(1010101, 1111010) = 5$$

$$d(1011010, 1111010) = 1$$

$$d(1100011, 1111010) = 3$$

$$d(1101100, 1111010) = 3$$

$$d(1110000, 1111010) = 2$$

$$d(1111111, 1111010) = 3$$

Observe que $d(1011010, 1111010) = 1$, que é justamente o valor de k para esse código. Pela distância de Hamming mínima foi possível detectar o erro e corrigir a mensagem recebida, que agora sabemos que originalmente era $[\ 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0\ ]$, observe o diagrama.

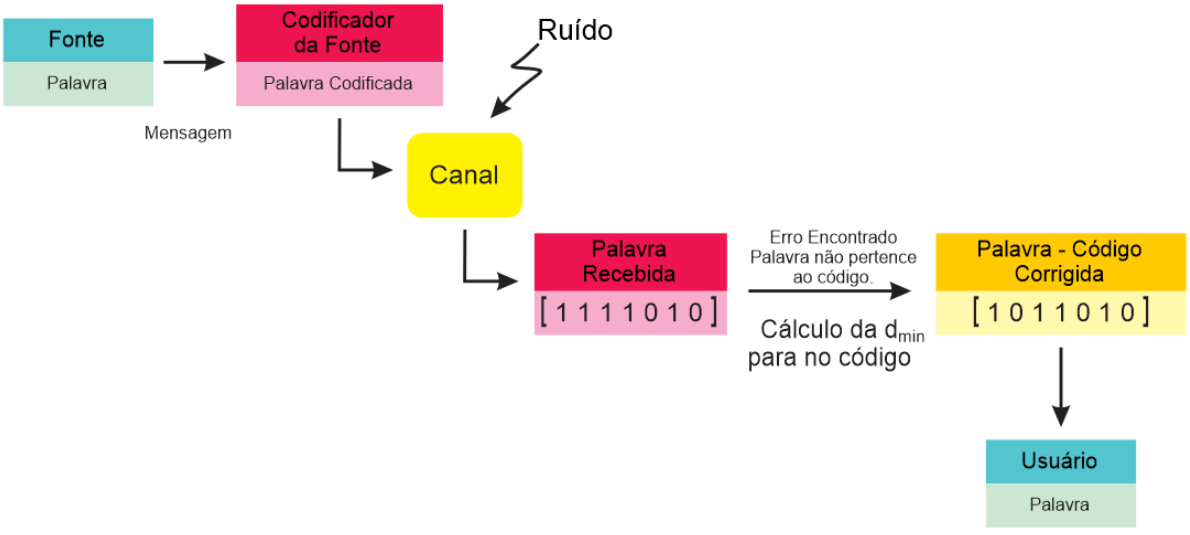


Figura 2.1: Correção pela distância mínima

A importância da *distância de Hamming mínima* de um código está no teorema a seguir.

Teorema 2.22. Seja um código $\mathcal{C}$ com distância de Hamming mínima d. Então:

(a) $\mathcal{C}$ detecta até k erros se, e somente se, $k \leq d-1$;

(b) $\mathcal{C}$ corrige até k erros se, e somente se, $2k \leq d-1$, ou seja, $k \leq \left\lfloor \frac{d-1}{2} \right\rfloor$.

Demonstração. (a) Existindo duas palavras de código $\mathbf{u}$ e $\mathbf{v}$ em $\mathcal{C}$ tais que $d(\mathbf{u},\mathbf{v}) = d$, se a palavra $\mathbf{u}$ for transmitida e acontecerem d erros que a transformem em $\mathbf{v}$, então esses erros nunca serão detectados. Portanto, se $\mathcal{C}$ detecta até k erros, então $k < d$.

Reciprocamente, suponhamos que na transmissão de uma palavra $\mathbf{u} \in \mathcal{C}$ ocorreram k erros, resultando na palavra $\mathbf{u}$.

Logo $d(\mathbf{u},\mathbf{v}) = k$.

Daí para provarmos que o código terá a capacidade de detectar o erro, teremos que garantir que $\mathbf{v} \notin \mathcal{C}$, o que é fácil, se a distância $d(\mathbf{u},\mathbf{v}) = k$ e $k < d$ (que é a distância mínima no código $\mathcal{C}$) e $\mathbf{u} \in \mathcal{C}$ então $\mathbf{v} \notin \mathcal{C}$.

(b) Se $\mathcal{C}$ corrige até k erros, então $2k \leq d-1$.

De fato, $d = 2k$ implicaria a existência de duas palavras $\mathbf{u}$ e $\mathbf{v}$ diferindo exatamente em $2k$ posições; acontecendo k erros em metade dessas $2k$ posições na transmissão de $\mathbf{u}$, nunca seria possível corrigir esses erros, pois poderia ter sido a palavra $\mathbf{v}$ a palavra emitida (tendo os k erros ocorrido na outra metade dessas $2k$ posições).

Reciprocamente, suponhamos que na transmissão de uma palavra $\mathbf{u} \in \mathcal{C}$ ocorreram k erros, resultando na palavra recebida $\mathbf{v}$ (portanto, $d(\mathbf{u},\mathbf{v}) = k$). Agora, para provarmos que o código terá a capacidade de corrigir o erro, bastará garantir que mais nenhuma palavra em $\mathcal{C}$ além de $\mathbf{u}$ pode ter dado origem à palavra errada $\mathbf{v}$, ou seja, que qualquer outra palavra $\mathbf{t} \in \mathcal{C}$ está a uma distância de $\mathbf{v}$ maior do que k. Isto resulta da desigualdade triangular da distância:

$d(\mathbf{u},\mathbf{t}) \leq d(\mathbf{u},\mathbf{v}) + d(\mathbf{v},\mathbf{t}) \Rightarrow d(\mathbf{v},\mathbf{t}) \geq d(\mathbf{u},\mathbf{t}) - d(\mathbf{u},\mathbf{v}) \geq d - k \geq 2k + 1 - k = k + 1$ □

Portanto, se quaisquer duas palavras do código estiverem a uma distância de Hamming, de pelo menos, $k+1$, um código consegue detectar k erros.

Exemplo 2.23. Suponha que uma mensagem seja um dígito binário simples: 0 ou 1. Se codificarmos a mensagem simplesmente repetindo-a duas vezes teremos as matrizes de código $\begin{bmatrix} 0 & 0 \end{bmatrix}$ e $\begin{bmatrix} 1 & 1 \end{bmatrix}$ com distância mínima $d = 2$.

Esse código poderá detectar erros em uma componente, porém não pode corrigi-lo, veja:

Se transmitirmos $[\ 0 \quad 0\ ]$ e um erro ocorrer na primeira componente, $[\ 1 \quad 0\ ]$ será recebida e o erro será detectado, porque essa não é uma matriz de código permitida. Porém o receptor não poderá corrigi o erro, já que $[\ 1 \quad 0\ ]$ também seria o resultado de erro na segunda componente se $[\ 1 \quad 1\ ]$ tivesse sido transmitida.

Exemplo 2.24. Suponha agora que a mensagem de dígito binário simples: 0 ou 1. Seja codificada repetindo-a três vezes, teremos as matrizes de código $[\ 0 \quad 0 \quad 0\ ]$ e $[\ 1 \quad 1 \quad 1\ ]$ com distância mínima $d = 3$.

Esse código tem a capacidade de detectar erros em uma componente e corrigi-lo nessa componente, veja:

Se $[\ 0 \quad 1 \quad 0\ ]$ for recebida, saberemos que dever ter sido resultado de um erro em uma componente na transmissão de $[\ 0 \quad 0 \quad 0\ ]$, já que um erro em uma componente de $[\ 1 \quad 1 \quad 1\ ]$ não a teria produzido.

Proposição 2.25. A distância de Hamming mínima de um código de paridade é 2.

Demonstração. Sejam $A = [\ a_1 \quad \cdots \quad a_i \quad \cdots \quad a_n \quad a_{n+1}\ ]$ e $B = [\ b_1 \quad \cdots \quad b_i \quad \cdots \quad b_n \quad b_{n+1}\ ]$ palavras de um código de paridade. Se $d(A, B) \leq 1$ então $A = B$, logo a distância mínima do código é maior ou igual a 2.

Como $[\ 0 \quad 0 \quad 0 \quad \cdots \quad 0\ ]$ $[\ 1 \quad 1 \quad 0 \quad \cdots \quad 0\ ]$ são palavras do código com distância de Hamming 2, isso completa a demonstração. □

A tabela a seguir mostra as capacidades de correção de erros dos códigos, dependendo da distância mínima.

Distância Mínima	1	2	3	4	5	6	7	8	9
Erros Detectados	0	1	2	3	4	5	6	7	8
Erros Corrigidos	0	0	1	1	2	2	3	3	4

Tabela 2.3: Capacidade de detecção e correção de erros dos códigos

2.4 Código de Hamming

Nesta seção usaremos as definições e demonstrações de acordo com [14].

No Exemplo 2.24 vimos que o código $\{[\ 0\quad 0\quad 0\ ], [\ 1\quad 1\quad 1\ ]\}$ tem capacidade de corrigir um erro em uma componente. Vejamos como este código pode ser descrito por meio de uma multiplicação de matrizes.

Seja $G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ (A matriz G é a *matriz geradora* do código). Faremos a transformação através do produto $G\mathbf{x}$, onde $\mathbf{x} \in \{[0], [1]\}$

Para saber se uma matriz recebida é uma matriz de código, é necessário que $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$ satisfaça $u_1 = u_2 = u_3$. Podemos escrever essas equações em um sistema linear sobre $\mathbb{Z}_2$:

$$\begin{array}{lll} u_1 = u_2 & & u_1 + u_2 \quad\quad = 0 \\ & \text{ou} & \\ u_1 = u_3 & & u_1 \quad\quad +u_3 = 0 \end{array} \tag{1}$$

Se $P = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$, o sistema (1) é equivalente a

$$P\mathbf{u} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{0}.$$

A matriz P é chamada de *matriz de verificação de paridade* para o código.

Observe que $PG = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+1+0 \\ 1+0+1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{0}$.

Para ver como essas matrizes ajudam na correção de erros, imagine que mandamos [1] como $[\ 1\quad 1\quad 1\ ]$, mas um erro em uma componente a faz ser recebida como $\mathbf{v} = [\ 1\quad 0\quad 1\ ]$. Calculando

$$P\mathbf{v} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \neq \mathbf{0}$$

Por isso sabemos que $\mathbf{v}$ não pode ser uma matriz de código.

Observe que a matriz de checagem determina se a matriz recebida é uma matriz de código, e ainda, caso ocorra um erro de envio ela determina onde ocorreu o erro, perceba que o erro

está na segunda componente da matriz recebida e o cálculo do produto de $P\mathbf{v} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ que é justamente a segunda coluna da matriz de verificação e onde ocorre o erro na matriz recebida.

Exemplo 2.26. Imagine novamente que mandamos [1] como $[\ 1 \quad 1 \quad 1\ ]$, mas um erro em uma componente a faz ser recebida como $\mathbf{v} = [\ 0 \quad 1 \quad 1\ ]$. Calculamos

$$P\mathbf{v} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0+1+0 \\ 0+0+1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \neq \mathbf{0}$$

Novamente vemos que a matriz recebida não é de código e a componente errada, segundo a matriz de verificação de paridade, é a primeira.

Definição 2.27. Sejam $n > k$ inteiros positivos e G uma matriz binária de ordem $n \times k$. O código $\mathcal{C} = \{G\mathbf{x} : \ x \in \mathbb{Z}_2^k\} \subseteq \mathbb{Z}_2^n$ é um *código binário* (n, k).

Definição 2.28. ([14]) Se $k < n$, toda matriz $n \times k$ da forma $G = \begin{bmatrix} I_k \\ A \end{bmatrix}$, em que A é uma matriz $(n-k) \times k$ sobre $\mathbb{Z}_2$, é chamada de *matriz geradora padrão* para um *código binário* (n, k). Qualquer matriz $(n-k) \times n$ da forma $P = [B \ I_{n-k}]$, em que B é uma matriz $(n-k) \times k$ sobre $\mathbb{Z}_2$, é chamada *matriz de verificação de paridade padrão*. Dizemos então que o código tem *comprimento* n e *dimensão* k.

Exemplo 2.29. No Exemplo 1.6 codificamos as mensagens *cima, baixo, esquerda, direita* conforme a tabela:

Mensagem	Cima	Baixo	Esquerda	Direita
Código	$[\ 0 \quad 0\ ]$	$[\ 0 \quad 1\ ]$	$[\ 1 \quad 0\ ]$	$[\ 1 \quad 1\ ]$

Tabela 2.4: Mensagem Codificada

Para projetar um código para enviar estas mensagens, uma candidata a matriz geradora G é:

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

E a matriz de verificação de paridade P é da forma:

$$P = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Fazendo a transformação pelo produto $G\mathbf{x}$, temos:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Assim temos as matrizes de código a serem enviadas, e se recebida alguma diferente das obtidas pela transformação pelo produto $G\mathbf{x}$, sabemos que a mensagem foi recebida com erro.

Observe que

$$PG = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \mathbf{0}$$

Daí, se ao enviar uma matriz de código $\mathbf{u}$ e a mesma for recebida como sendo $[\ 1 \quad 0 \quad 0 \quad 1 \quad 1\ ]$, vemos que ouve um erro (não é uma das possíveis), mas qual era a mensagem original? Basta usar a matriz de paridade para verificar onde se encontra o erro.

$$P\mathbf{v} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+0+0+0+0 \\ 0+0+0+1+0 \\ 1+0+0+0+1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \neq \mathbf{0}$$

Confirmamos pela matriz de verificação de paridade que a matriz recebida não é de código e ainda sabemos que o erro está na segunda componente.

Portanto a matriz de código correta é [1 1 0 1 1], que decodificada é a mensagem [1 1], portanto *Direita*.

Observe na Figura 2.2 a descrição gráfica do Exemplo 2.29.

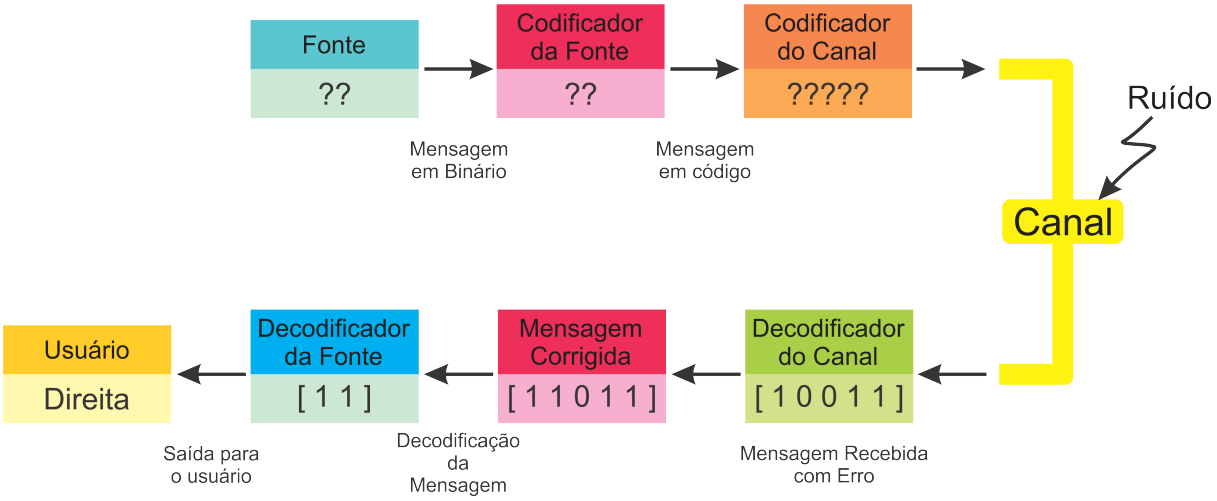


Figura 2.2: Esquema de codificação e decodificação

Teorema 2.30. Se $G = \begin{bmatrix} I_k \\ A \end{bmatrix}$ for uma matriz geradora padrão e $P = [B\ I_{n-k}]$ for uma matriz de verificação de paridade padrão, P será a matriz de verificação de paridade associada a G se, e somente se, $A = B$. O código binário correspondente (n, k) será um corretor de erros (em uma componente) se, e somente se, as colunas de P forem não nulas e distintas.

Demonstração. Sendo P e G como na hipótese do teorema, assuma primeiramente que elas sejam matriz de verificação de paridade padrão e matrizes geradoras do mesmo código binário. Portanto para todo $\mathbf{x}$ em $\mathbb{Z}_2^k$, $PG\mathbf{x} = \mathbf{0}$. Usando a multiplicação por blocos, temos:

$$\begin{bmatrix} B & I \end{bmatrix} \begin{bmatrix} I \\ A \end{bmatrix} \mathbf{x} = \begin{bmatrix} BI + IA \end{bmatrix} \mathbf{x} = 0\mathbf{x} = 0 \text{ para todo } \mathbf{x} \text{ em } \mathbb{Z}_2^k$$

Equivalentemente, para todo $\mathbf{x}$ em $\mathbb{Z}_2^k$, temos:

$$B\mathbf{x} + A\mathbf{x} = (B + A)\mathbf{x} = (BI + IA)\mathbf{x} = \begin{bmatrix} B & I \end{bmatrix} \begin{bmatrix} I \\ A \end{bmatrix} \mathbf{x} = 0$$

ou

$$B\mathbf{x} = A\mathbf{x}$$

Se agora tomarmos $\mathbf{x} = \mathbf{e}_i$ o i-ésima matriz de $\mathbb{Z}_2^k$, vemos que:

$\mathbf{b}_i = B\mathbf{e}_i = A\mathbf{e}_i = \mathbf{a}_i$ para todo i, onde $\mathbf{a}_i$ é a i-ésima coluna de uma matriz A.

Portanto $B = A$.

Reciprocamente, é fácil verificar que, se $B = A$, $PG\mathbf{x} = 0$ para todo $\mathbf{x}$ em $\mathbb{Z}_2^k$.

Para provar que esse par determina um código corretor de erros se as colunas P forem não nulas e distintas, considere $\mathbf{x}$ como uma matriz de mensagem em $\mathbb{Z}_2^k$ e $\mathbf{u} = G\mathbf{x}$ como a matriz de código correspondente. Então, $P\mathbf{u} = 0$. Se houvesse um erro na i-ésima componente, resultando em uma matriz $\mathbf{v}$, teríamos $\mathbf{v} = \mathbf{u} + \mathbf{e}_i$.

Calculamos agora

$$P\mathbf{v} = P(\mathbf{u} + \mathbf{e}_i) = P\mathbf{u} + P\mathbf{e}_i = \mathbf{0} + \mathbf{p}_i = \mathbf{p}_i$$

que aponta o erro na i-ésima componente.

Por outro lado, se $\mathbf{p}_i = \mathbf{0}$, um erro na i-ésima componente não será detectado (por exemplo $P\mathbf{v} = \mathbf{0}$) , e, se $\mathbf{p}_i = \mathbf{p}_j$, não poderemos determinar se um erro ocorreu na i-ésima ou na j-ésima componente. □

Exemplo 2.31. Ao desenvolver um código corretor de erros, pretendemos que ele seja o maior possível para que possamos transmitir a maior quantidade de informações possível. Considere a ideia de criar um código corretor de erros que use três equações de verificação de paridade, dessas equações formam-se as linhas de P. Como teremos três linhas na matriz de verificação de paridade e pelo teorema anterior, as n colunas de P precisam ser não nulas e distintas, o número máximo de colunas ocorre em $2^3 - 1 = 7$ casos. Uma candidata a pode ser

$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Isso significa que

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

e, dessa forma, pelo teorema, uma matriz geradora padrão para esse código é

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Para exemplificar como a matriz geradora funciona, suponha a codificação de $\mathbf{x} = [\ 0 \quad 1 \quad 0 \quad 1\ ]$ para obtermos a matriz de código

$$\mathbf{u} = G\mathbf{x} = [\ 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0\ ]$$

Se recebido, essa matriz será considerada como correta, já que $P\mathbf{u} = \mathbf{0}$.

Por outro lado, se $\mathbf{v} = [\ 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0\ ]$ for recebida, calculamos

$$P\mathbf{v} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

que reconhecemos como a terceira coluna de P. Portanto, o erro está na terceira componente de $\mathbf{v}$, e, alterando essa componente, recuperamos a matriz de código correta $\mathbf{u}$. Sabemos também que as quatro primeiras componentes de uma matriz de código são a matriz de mensagem original; logo, neste caso, decodificamos $\mathbf{u}$ e obtemos a matriz original $\mathbf{x} = [\ 0 \quad 1 \quad 0 \quad 1\ ]$.

Exemplo 2.32. Vamos tomar outro exemplo utilizando as mesmas matrizes P e G do exemplo anterior. Suponha a codificação da matriz $\mathbf{y} = [\ 1 \quad 1 \quad 0 \quad 1\ ]$ para obtermos a matriz de código

$$\mathbf{u} = G\mathbf{y} = [\ 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0\ ]$$

Se recebido, essa matriz será considerada como correta, já que $P\mathbf{u} = 0$.

Por outro lado, se $\mathbf{v} = [\ 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0\ ]$ for recebida, calculamos

$$P\mathbf{v} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

novamente reconhecemos a coluna de P onde está o erro, e, alterando essa componente, recuperamos a matriz de código correta $\mathbf{u}$. Logo obtemos a matriz original $\mathbf{y} = [\ 1 \quad 1 \quad 0 \quad 1\ ]$.

O código utilizado nesses exemplos é chamado código de Hamming $(7, 4)$. Qualquer código binário construído dessa forma é chamado *código de Hamming* (n, k).

2.4.1 Um Truque de Mágica Baseado no Código de Hamming

Voltando ao truque apresentado no Capítulo 1, o que aconteceria se o voluntário mentisse? Evidentemente o Mágico não acertaria o número pensado, já que o truque é baseado em um código (informação sobre onde está o número pensado) transmitido de forma correta.

Os passos descritos a seguir foram construídos baseados em [3], [5], [10] e [13].

Agora, suponha que o mago permita ao voluntário mentir no máximo uma vez. Nesse caso teríamos uma mensagem transmitida com 1 erro (mentira). Utilizando o código de Hamming $(7, 4)$ podemos encontrar onde está o erro e decodificar a mensagem, ou seja, saber em qual cartão ele mentiu e ainda qual o número inicialmente pensado. Vamos ao truque:

Utilizando a Matriz Geradora G apresentada no Exemplo 2.31 e codificando os números de 1 a 15 temos:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Número Escolhido ($\mathbf{x}$)		$\mathbf{x}$ · Binário		$G\mathbf{x}$ = Código $(7,4)$	Peso de $(G\mathbf{x})$
1	$\rightarrow$	0001	$\rightarrow$	0001111	3
2	$\rightarrow$	0010	$\rightarrow$	0010011	3
3	$\rightarrow$	0011	$\rightarrow$	0011100	3
4	$\rightarrow$	0100	$\rightarrow$	0100101	3
5	$\rightarrow$	0101	$\rightarrow$	0101010	3
6	$\rightarrow$	0110	$\rightarrow$	0110110	4
7	$\rightarrow$	0111	$\rightarrow$	0111001	4
8	$\rightarrow$	1000	$\rightarrow$	1000110	3
9	$\rightarrow$	1001	$\rightarrow$	1001001	3
10	$\rightarrow$	1010	$\rightarrow$	1010101	4
11	$\rightarrow$	1011	$\rightarrow$	1011010	4
12	$\rightarrow$	1100	$\rightarrow$	1100011	4
13	$\rightarrow$	1101	$\rightarrow$	1101100	4
14	$\rightarrow$	1110	$\rightarrow$	1110000	3
15	$\rightarrow$	1111	$\rightarrow$	1111111	7

Agora que transformamos cada número $\mathbf{x}$ em um código, podemos detectar qualquer erro simples que aconteça e ainda corrigí-lo.

Para confeccionar os novos cartões usaremos o método já utilizado: Trocando cada 1 e 0, por S (SIM) e N (NÃO), temos em quais cartões cada número deverá aparecer ou não:

$$1 \rightarrow 0001111 \rightarrow NNNSSSS$$

$$2 \rightarrow 0010011 \rightarrow NNSNNSS$$

$$3 \rightarrow 0011100 \rightarrow NNSSSNN$$

$$4 \rightarrow 0100101 \rightarrow NSNNSNS$$

$$5 \rightarrow 0101010 \rightarrow NSNSNSN$$

$$6 \rightarrow 0110110 \rightarrow NSSNSSN$$

$$7 \rightarrow 0111001 \rightarrow NSSSNNS$$

$$8 \rightarrow 1000110 \rightarrow SNNNSSN$$

$$9 \rightarrow 1001001 \rightarrow SNNSNNS$$

$$10 \rightarrow 1010101 \rightarrow SNSNSNS$$

$$11 \rightarrow 1011010 \rightarrow SNSSNSN$$

$$12 \rightarrow 1100011 \rightarrow SSNNNSS$$

$$13 \rightarrow 1101100 \rightarrow SSNSSNN$$

$$14 \rightarrow 1110000 \rightarrow SSSNNNN$$

$$15 \rightarrow 1111111 \rightarrow SSSSSSS$$

Portanto os cartões serão:

8	9	10	11
12	13	14	15

4	5	6	7
12	13	14	15

2	3	6	7
10	11	14	15

1	3	5	7
9	11	13	15

1	3	4	6
8	10	13	15

1	2	5	6
8	11	12	15

1	2	4	7
9	10	12	15

Porém, se você for jogar assim não terá êxito da mesma forma, pois serão SIMs e NÃOs como resposta para o número se encontrar ou não no cartão apresentado, além da possível mentira do

voluntário. Portanto terá que usar a matriz

$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

de verificação de paridade apresentada no Exemplo 2.31, para decodificar e descobrir em qual coluna dela está o erro, assim resolver o problema e achar o número pensado, o que não é nada excitante do ponto de vista de um truque. Veja no exemplo uma situação hipotética:

Exemplo 2.33. Imagime que tenhamos a resposta NSSNNNS, qual o número pensado pelo voluntário?

Calculando

$$P\mathbf{v} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0+1+0+0+0+0+0 \\ 0+0+1+0+0+0+0 \\ 0+1+1+0+0+0+1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Pela matriz P, temos que o erro está na quarta componente, corrigindo-a, descobrimos a matriz de código correta. Utilizando somente suas 4 primeiras componentes, temos: $\begin{bmatrix} 0 & 1 & 1 & 1 \end{bmatrix}$. Após decodificado temos o número 7 como o número pensado pelo voluntário.

Agora, o que fazer para melhorar a dinâmica do jogo? Simples, vamos melhorar nossos cartões, a fim de que a resposta correta seja mostrada por eles próprios.

Ao invés de usarmos os cartões no formato da página anterior, usaremos cartões quadrados e com abas ao seu redor, cada aba representando um dos possíveis números no jogo, incluindo o zero.

Esse cartão será o cartão-base (Figura 2.3), sobre o qual o truque acontecerá e a seguir os sete cartões-pergunta, que serão conforme a Figura 2.4.

Observe como o conjunto dos números de cada cartão são complementos das abas que aparecem no mesmo cartão. Veja: O cartão Azul é composto pelos números 2, 3, 6, 7, 10, 11, 14 e

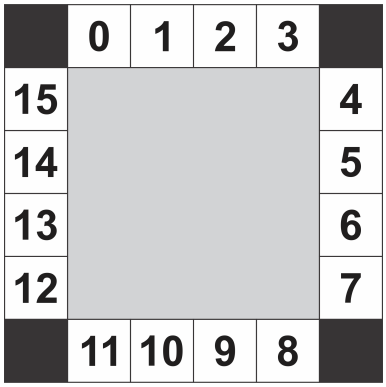


Figura 2.3: Cartão-base

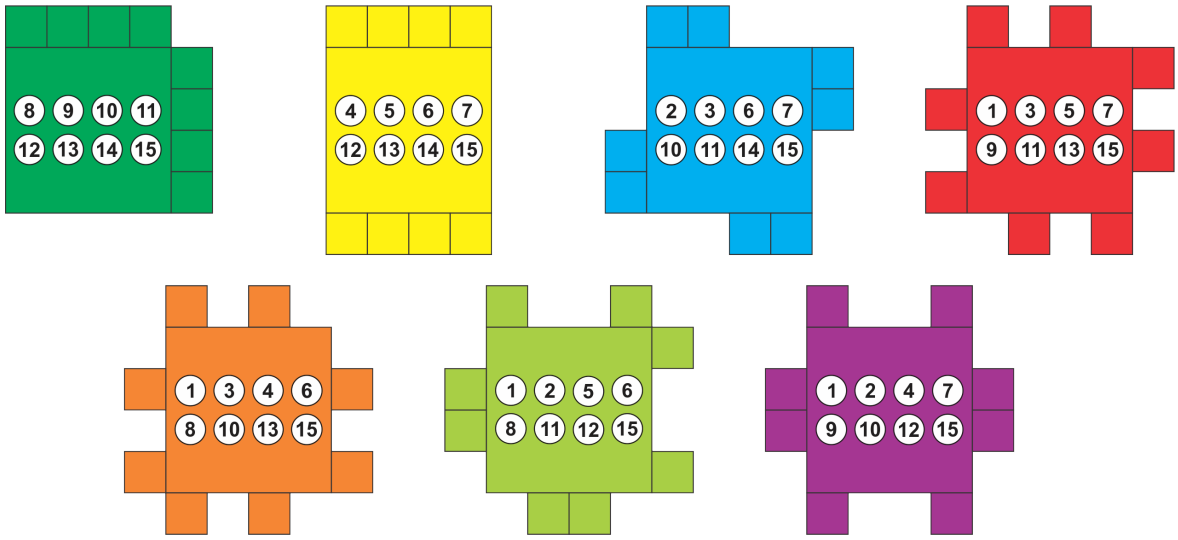


Figura 2.4: Cartões-pergunta

15. Logo as abas cobrirão os seus complementares (aqueles que não estão no cartão) 0, 1, 4, 5, 8, 9, 12 e 13. Veja na Figura 2.5.

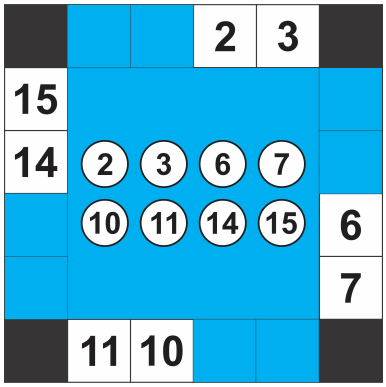


Figura 2.5: Cartão Azul - exemplo das abas

Cada cartão codificará uma pergunta. Sempre que a resposta é SIM o cartão é mantido com o lado direito para cima, se a resposta dada é NÃO o cartão é invertido através de uma diagonal

do canto superior esquerdo para o canto inferior direito, como mostrado na Figura 2.6. Observe que o cartão na sua orientação original será revelado no estado invertido, e vice-versa.

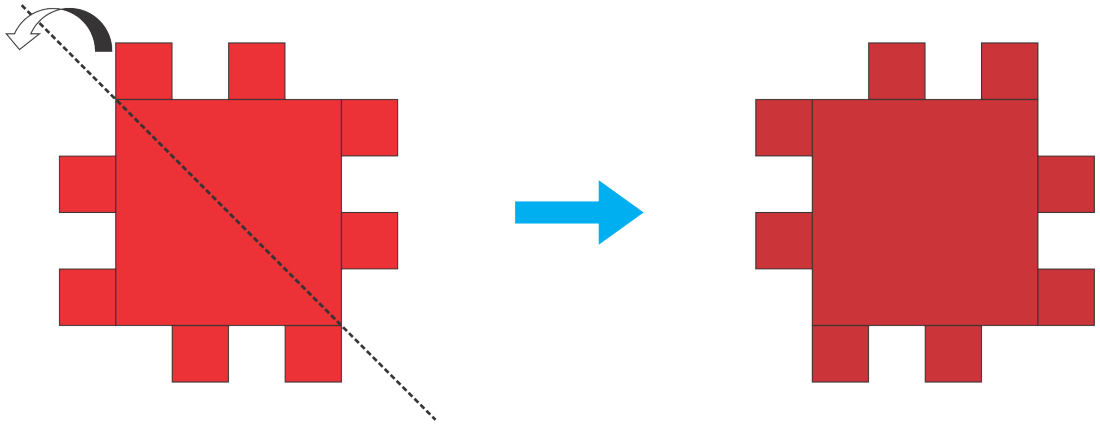


Figura 2.6: Virar Cartão

Exemplo 2.34. Vamos exemplificar uma situação de jogo:

Supondo que o Voluntário escolha o número 12 e ao ver os cartões responda, nessa ordem, $SSNSNSS$, o Mágico colocará os cartões e terá as seguintes figuras na ordem em que aparecem:

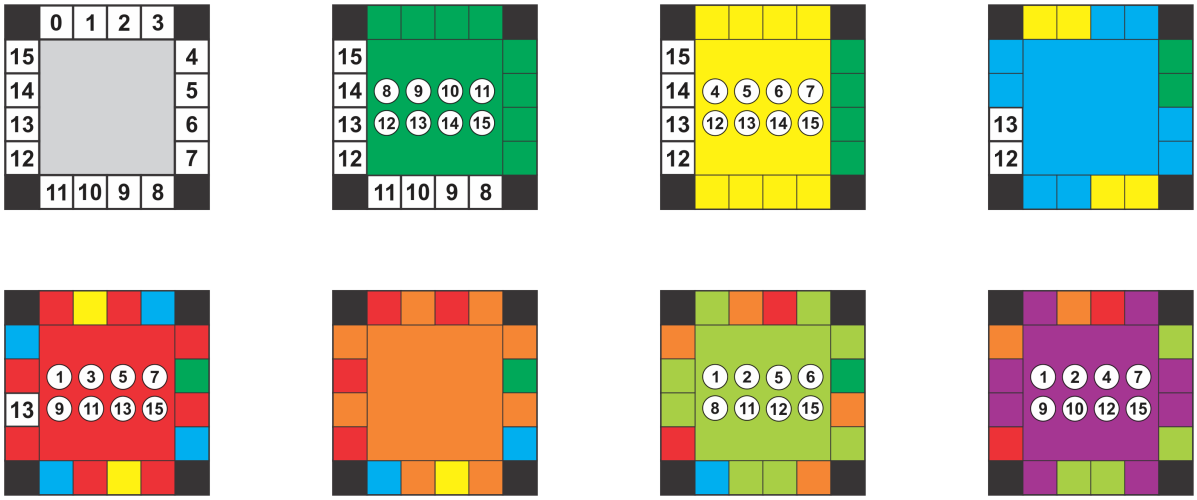


Figura 2.7: Situação de Jogo

Mas qual é o número pensado? Onde está a resposta? Na teoria é fácil, é só procurar o número que está coberto por um único cartão, tarefa que na prática, não é tão simples; não há maneira de determinar este local se sem manusear os cartões, o que faz o truque menos impressionante. Analisando cor a cor: todas as abas verdes foram cobertas, as abas amarelas também foram cobertas, logo o número está de 12 a 15, inspecionando novamente, as abas azuis referentes ao 14 e 15, também foram cobertas, sobraram somente o 12 e 13; o número 13 só foi coberto a primeira vez pela cor alaranjada, que foi coberta pelo verde oliva e logo após pelo roxo,

sobrando o número 12 que só foi coberto pela cor vermelha, portanto, o número escolhido pelo voluntário é o 12, o que confirma a escolha do exemplo.

Em tempo, analisando as respostas do Voluntário, foram: $SSNSNSS$, o que mostra que ele mentiu no 4º cartão, o cartão Vermelho, novamente o código de Hamming se mostra detector e corretor de erros, conforme já foi mostrado, pois o cartão que cobriu o 12 era o vermelho.

Agora, a dinâmica do truque fica prejudicada por essa demora em achar qual o número que foi encoberto somente uma vez, dependendo do olhar apurado do Mágico e/ou manuseio dos cartões. Para implementar a identificação do número e superar a dificuldade fazemos pequenos furos nas abas de cada cartão, permitindo que o cartão-resposta fique facilmente em evidência quando coberto por apenas uma aba. Na Figura 2.8 temos as possíveis posições do furo nas abas.

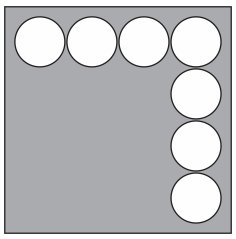


Figura 2.8: Posições dos furos nas abas

Cada cartão-pergunta tem de ter a sua própria posição do furo nas abas, e o mesmo deve ser colocado nesta mesma posição relativa, dois cartões não podem compartilhar a mesma posição dos furos, mesmo se um dos cartões está invertido. Um exemplo dos furos nos cartões é (Figura 2.9):

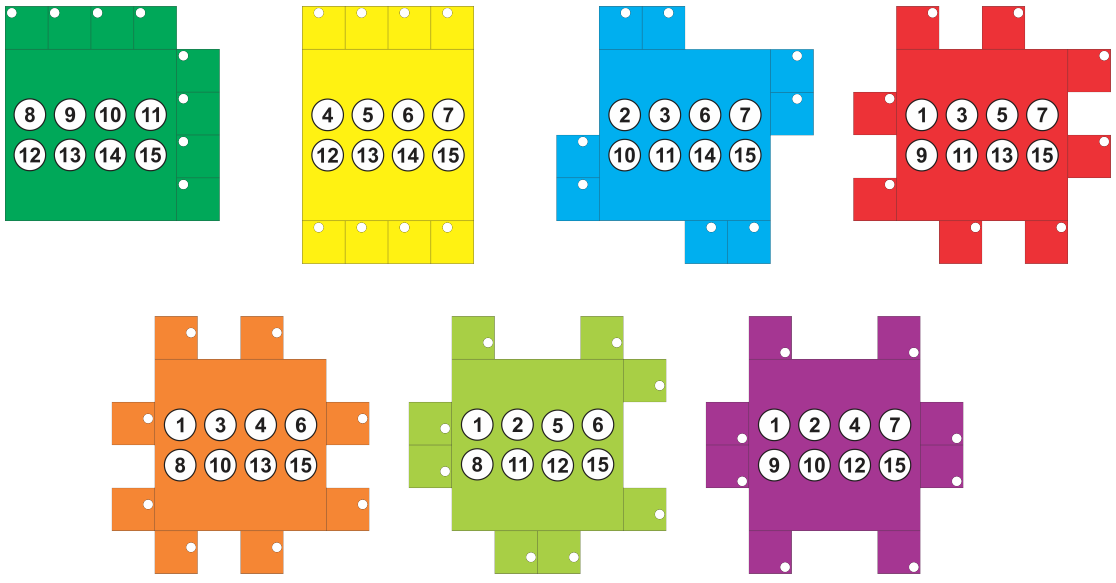


Figura 2.9: Cartões com as abas e furos

Agora, refazendo o exemplo anterior com os novos cartões, temos:

Exemplo 2.35. O Voluntário pensa no número 12 e responde, $SSNSNSS$, o Mágico coloca os cartões conforme as respostas e tem as seguintes figuras na ordem em que aparecem:

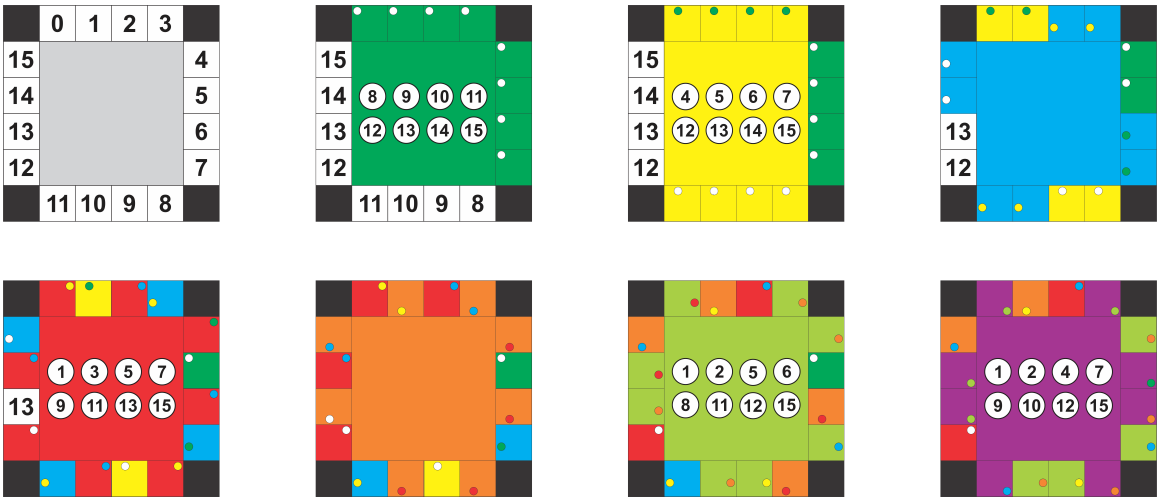


Figura 2.10: Truque com novos cartões

Olhemos mais de perto o último quadro (Figura 2.11). Após aplicação do presente regime,

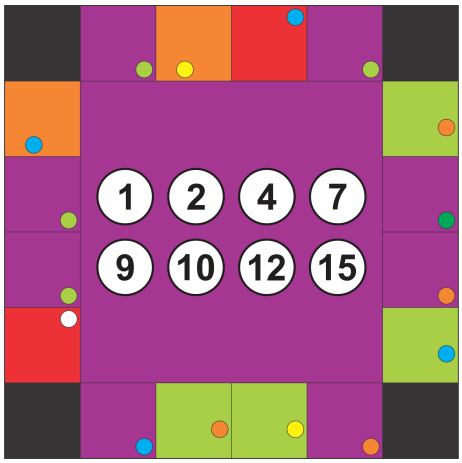


Figura 2.11: Truque com novos cartões

o Mago vai observar um único buraco branco na conclusão do truque. A localização do orifício branco nos dá o número selecionado pelo voluntário, além da cor da aba do cartão em que o voluntário mentiu. Aqui, vemos um buraco branco na posição em torno do número 12 e a cor vermelha. Então o mágico identifica este número e cor como a seleção do voluntário.

Para aprender este truque mágico pode-se usar um cartão de base com os 15 números exibidos. Eventualmente, um cartão de base sem que os números podem ser usados de modo que seja mais difícil para o público descobrir o que está acontecendo. O mago simplesmente precisa memorizar

as localizações dos números do cartão de base. Isso não deve ser difícil, já que os números aparecem em uma ordem no sentido horário a partir do canto superior esquerdo.

Um caso muito interessante é se o Voluntário for sincero em todos os cartões. Caso isso aconteça, após colocar o último cartão, somente um número estará à vista.

Exemplo 2.36. Vejamos em um novo exemplo: Se o voluntário pensar no número 14 e for sincero em todas as respostas teremos a sequência $SSSNNNN$, e a sequência de cartões será:

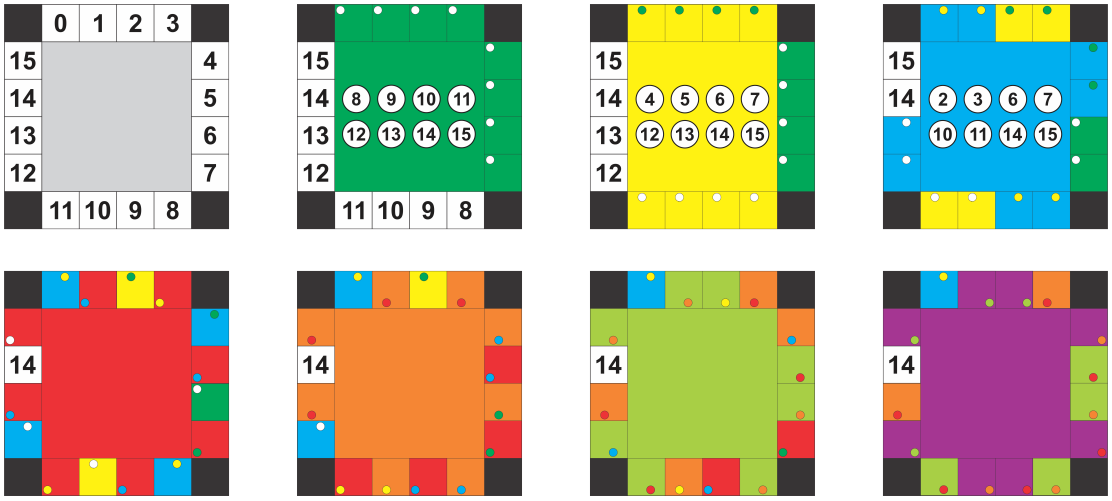


Figura 2.12: Truque com novos cartões

O código de Hamming decodifica a mensagem verdadeira, a sequência de respostas é o código transmito, e como não houve ruído (mentira) em sua transmissão, o mesmo já foi decodificado corretamente.

Capítulo 3

Modelos de Atividades Propostas

Números de adivinhações aritméticas têm sido apresentados a pessoas e alunos de vários níveis de escolaridade e sempre causam surpresa e fazem muito sucesso [2].

Atualmente existe uma enorme utilização do sistema decimal de numeração, seja na hora de falar, seja para representar quantidades, porém, o que não é muito conhecido é o papel fundamental do sistema binário em nossas vidas. Esse sistema é a base para o modo de armazenamento e processamento de informações nos computadores [4].

Esta atividade faz esse elo *decimal-binário*, apresentando conteúdos através de uma mágica com cartelas que será apresentada aos alunos e o desafio será entender como a mágica funciona. Finalizando fixaremos o foco nessa aplicação do sistema binário. A atividade é direcionada a duas etapas de ensino: a primeira, aos alunos das séries finais do Ensino Fundamental (8º e 9º Ano); a segunda aos alunos que estejam cursando, no mínimo, o segundo ano do ensino médio, uma vez que já terão visto o conteúdo de matrizes.

A finalidade dessa atividade é fazer com que a matemática tenha maior aplicabilidade no dia-a-dia dos alunos. Com essa proposta, espera-se que os mesmos consigam associar o conteúdo visto em sala de aula com situações do seu cotidiano.

Os alunos inicialmente veem o truque como algo desligado da realidade, mas após, no fechamento espera-se que a turma tenha uma boa noção da relação entre esses tópicos: sistema binário, matrizes, transmissão e armazenamento de informação; aparentemente não relacionados com a realidade matemática vista em sala de aula.

3.1 Um Truque com Números Binários

O truque aqui apresentado baseia-se somente na representação binária dos números naturais, assim como em [5], e é direcionado principalmente aos alunos das séries finais do Ensino Fundamental, podendo ser trabalhado com os 4 cartões (de 1 a 15), com os 5 cartões (de 1 a 31) ou ainda com 6 cartões (de 1 a 63).

8	9	10	11
12	13	14	15

4	5	6	7
12	13	14	15

2	3	6	7
10	11	14	15

1	3	5	7
9	11	13	15

Figura 3.1: Cartões com números de 1 a 15.

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

Figura 3.2: Cartões com números de 1 a 31.

32	33	34	35	36	37
38	39	40	41	42	43
44	45	46	47	48	49
50	51	52	53	54	55
56	57	58	59	60	61
62	63				

16	17	18	19	20	21
22	23	24	25	26	27
28	29	30	31	48	49
50	51	52	53	54	55
56	57	58	59	60	61
62	63				

8	9	10	11	12	13
14	15	24	25	26	27
28	29	30	31	40	41
42	43	44	45	46	47
56	57	58	59	60	61
62	63				

4	5	6	7	12	13
14	15	20	21	22	23
28	29	30	31	36	37
38	39	44	45	46	47
52	53	54	55	60	61
62	63				

2	3	6	7	10	11
14	15	18	19	22	23
26	27	30	31	34	35
38	39	42	43	46	47
50	51	54	55	58	59
62	63				

1	3	5	7	9	11
13	15	17	19	21	23
25	27	29	31	33	35
37	39	41	43	45	47
49	51	53	55	57	59
61	63				

Figura 3.3: Cartões com números de 1 a 63.

Conteúdos

Potências;

Sistemas de numeração;

Bases numéricas;

Base Binária;

Divisibilidade.

Objetivos

Relembrar diferentes sistemas de numeração;

Aprofundar o estudo sobre a base binária;

Conhecer aplicações da base binária.

Duração

Duas aulas.

Material necessário

Lápis;

Borracha;

Folhas de papel A4 (Cartões do anexo).

A mágica - Procedimentos Metodológicos

1ª Etapa

O professor chega na sala mostra aos alunos os cartões impressos com números de 1 a 15 e em seguida pede a um deles que pense em um número e diga em quais cartões o número se encontra. Após descobrir o número deve haver um certo espanto pelos alunos.

O professor repete a atividade com outros alunos para chamar a atenção ao jogo. Para a atividade se tornar mais interessante, troca-se os cartões, usando agora cartões com números até 31 e repete o procedimento anterior.

2ª Etapa

Como nesse momento os alunos devem estar ansiosos para saber como ele advinha, o professor explica como o truque usando os seguintes procedimentos:

1º - Separe os alunos em grupos de 3 integrantes e distribua os cartões numerados de 1 a 31 aos grupos. (*Observação*: ideal para a realização do experimento é levar as cartelas prontas para os alunos, já que sua confecção gasta muito tempo e pode dar dicas sobre o mistério da mágica);

2º - Peça para que um dos alunos do grupo escolha um número entre 1 e 31;

3º - A seguir, solicite que ele separe os cartões em que o valor escolhido está presente, sem revelar tal número;

4º - Agora diga aos grupos que somem o primeiro valor do canto superior esquerdo de cada um dos cartões que ele escolheu inicialmente (Espera-se que haja um pequeno espanto em ver que essa soma dê o valor pensado pelo voluntário).

5º - Repita o procedimento e os alunos verão que o "truque" sempre funciona.

3ª Etapa

Nesse momento pode-se direcionar algumas perguntas aos alunos: *Qual é o truque? Por que o truque funciona?*

E então o grupo deve discutir e tentar explicar por que esse procedimento funciona. Esperamos que os alunos notem que a primeira célula de cada cartela é uma potência de 2. Os outros valores que compõem cada cartela são aqueles que apresentam, em sua representação binária, a mesma potência de 2 que aparece na sua primeira célula.

Sugira que cada grupo escreva uma tabela com as somas realizadas durante o truque para buscar as informações que desvendem o mistério. Caso ache necessário, peça também para escreverem as parcelas na sua forma fatorada, conforme o exemplo:

Número Escolhido	Soma dos números iniciais	Soma das potências de base 2
29	16+8+4+1	$2^4 + 2^3 + 2^2 + 2^0$
24	16+8	$2^4 + 2^3$
15	8+4+2+1	$2^3 + 2^2 + 2^1 + 2^0$
10	8+2	$2^3 + 2^1$

Tabela 3.1: Exemplos

Os grupos devem formular hipóteses sobre a explicação da mágica. Essas hipóteses podem ser apresentadas para a turma e uma discussão decide as que são válidas e as que não são. Por fim, introduza a notação da solução da mágica com potências 2 caso os grupos ainda não o tenham

feito e mostre que "todo número natural pode ser escrito como a soma de diversas potências distintas de 2" (não há necessidade de provar esse resultado).

4ª Etapa

Propor aos alunos que confeccionem novos cartões, agora com números de 1 até 63.

Questão para os alunos: Como ficariam os cartões se quiséssemos que os valores de adivinhação possíveis se estendessem até 63? Ou ainda como ficariam os cartões se fossem usados valores de 1 até 100.

Questão para os alunos: Qual é a relação entre a quantidade de cartelas e o valor máximo que elas apresentam?

Note que para cada expoente das potências de 2 é necessário um cartão. Assim, é possível fazer a generalização, já exposta no experimento:

Considerando que o último cartão terá como primeiro valor a potência 2^n, então teremos $n+1$ cartões, iniciados com $2^0, 2^1, 2^2, \cdots, 2^{n-1}, 2^n$ e, para escrever o máximo de valores sem atingir a potência seguinte, podemos escrever até o número $2^{n+1}-1$ (que é um valor a menos que a próxima potência de 2).

Fechamento

Para o fechamento da atividade apresentada, separamos um método para obtenção da representação na base binária de um valor originalmente na base decimal.

Comente com a turma sobre nosso sistema de numeração, o decimal, que utiliza apenas 10 algarismos para representar qualquer valor inteiro positivo e, na escrita desses valores, a posição que cada algarismos ocupa no número altera seu valor, ou seja:

$$1498 \neq 9841$$

Sistemas com essa propriedade são chamados *sistemas posicionais*.

Fazendo a decomposição dos números acima através de somas, temos:

$$\begin{aligned} 1985 &= 1000+900+80+5 &&= 1\times 10^3+4\times 10^2+9\times 10^1+8\times 10^0 \\ 8591 &= 8000+500+90+1 &&= 8\times 10^3+5\times 10^2+9\times 10^1+1\times 10^0 \end{aligned}$$

De maneira geral temos: $a, b \in \mathbb{N}$, com $b>1$, existem números naturais $r_0, r_1, r_2, \cdots, r_n < b$,

univocamente determinados, tais que

$$a = r_n b^n + r_{n-1} b^{n-1} + \cdots + r_2 b^2 + r_1 b^1 + r_0 b^0$$

Mostre exemplos de representação binária (base 2) e que nesse sistema os algarismos são apenas zeros e uns, mas a formação permanece a mesma. Por exemplo:

$$1011 \quad = \quad 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \quad = \quad 8 + 0 + 2 + 1 \quad = \quad 11$$

este valor é 11 na base dez.

Questão para os alunos: Como converter um número na base decimal para a base binária?

Resposta: Para converter um número da base decimal para a base binária, basta realizar divisões sucessivas por 2; os restos dessa divisão fornecerão os algarismos do número na nova base.

Exemplo 3.1. Vamos converter 75_{10} para base binária:

$$\begin{aligned} 75/2 &= 37, \textit{ resto } 1 \\ 37/2 &= 18, \textit{ resto } 1 \\ 18/2 &= 9, \textit{ resto } 0 \\ 9/2 &= 4, \textit{ resto } 1 \\ 4/2 &= 2, \textit{ resto } 0 \\ 2/2 &= 1, \textit{ resto } 0 \\ 1/2 &= 0, \textit{ resto } 1 \end{aligned}$$

Portanto, $75_{10} = 1001011_2$

Faça mais exemplos e assim os alunos poderão confeccionar novos cartões de 1 a 63, de 1 a 100, ou até o valor pretendido.

3.2 Um Truque com Paridade

Aqui veremos um truque de adivinhação, que se encontra em [1], que aplica de forma recreativa o conceito de par e ímpar. Esta atividade utiliza um truque de mágica para mostrar como detectar e corrigir erros quando dados foram corrompidos.

A motivação é ensinar às crianças um truque de "mágica", mas o objetivo é a paridade (par e ímpar).

Conteúdos

Cálculo Mental na adição de Números Naturais;

Paridade (par e ímpar).

Objetivos

1. Relembrar os conceitos de paridade (par e ímpar);
2. Conhecer aplicações de paridade.

Duração

Uma aula.

Material necessário

Cartões coloridos, baralho, moedas, etc.

O truque

Desde a primeira execução do truque as crianças se mostram interessadas em aprendê-lo, e, ao final você se oferece para mostrar-lhes como fazê-lo. Como em qualquer truque de mágica, uma certa quantidade de drama é útil para tornar a apresentação eficaz.

O truque requer uma pilha de cartas idênticas de dois lados. As cartas podem ser vermelhas de um lado e brancas no outro, por exemplo. Uma maneira fácil de fazê-las é cortar uma grande folha de papel-cartão que é colorida em um lado apenas.

Figura 3.4: Papel-cartão

Um pacote de cartas de baralho também é adequado.

Para a demonstração é mais fácil se você tem um conjunto de cartas com ímãs sobre eles. É possível comprar tiras de imã de geladeira. Você pode então colocar as cartas verticalmente em

Figura 3.5: Baralho

uma placa de metal (por exemplo, um quadro branco), que é fácil para a classe ver. Se a turma for pequena, as cartas podem ser colocadas no chão na frente das crianças postas em círculo para o truque ser executado.

Etapas do truque

1. Peça a uma ou duas crianças que coloquem as cartas para você na placa magnética, em forma retangular. Qualquer tamanho de retângulo é adequado, mas cerca de 5x5 cartas é bom. (Quanto maior o retângulo de cartas, mais impressionante o truque.) As crianças podem decidir aleatoriamente qual a posição para cada carta. Veja um exemplo de retângulo aleatório de 5x5 cartas.

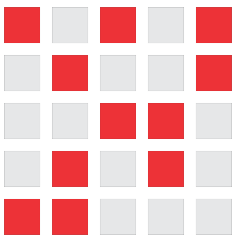


Figura 3.6: Retângulo 5x5 - Inicial

2. Casualmente, adicione outra linha e coluna ao retângulo "apenas para torná-lo um pouco mais difícil" (Figura 3.7). Claro, essas cartas são a chave para o truque. A estratégia é escolher as cartas extras para garantir que haja um número par de cartas coloridas em cada linha e coluna.

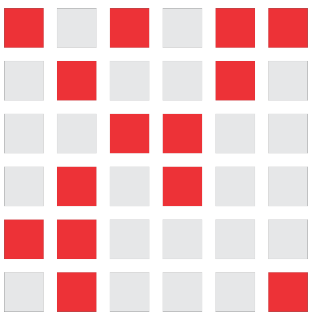


Figura 3.7: Retângulo 6x6 - Com os cartas extras

3. Selecione uma criança e, enquanto você cobrir seus olhos e/ou desviar o olhar, peça para a criança virar uma carta - apenas uma carta. Na Figura 3.8 temos um exemplo, a quarta carta na quarta linha foi invertida.

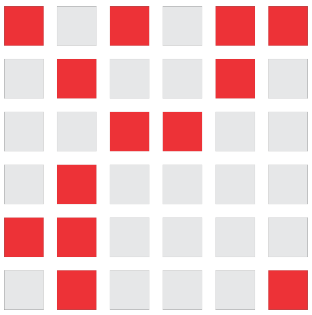


Figura 3.8: Retângulo 6x6 - Com uma carta virada

Em seguida você descobre seus olhos, estuda as cartas e identifica qual delas foi virada.

Como adivinhar?

Simples! Devido à forma como as cartas foram colocadas, a linha e a coluna que contém a carta alterada terão agora um número ímpar de cartas coloridas, o que identifica rapidamente a carta virada.

Agora, vire o carta de volta, e peça a outras crianças escolherem uma carta para virar. As crianças provavelmente estarão impressionadas com a sua capacidade de encontrar repetidamente a carta virada. O truque funciona com qualquer número de cartas, mas é muito mais impressionante se um monte de cartas são usadas, embora seja importante ensaiar o truque neste caso.

4. Peça às crianças que tentem adivinhar como o truque é feito. Este é um bom exercício de raciocínio, e também ajuda a ver que o método não é óbvio.

5. Neste ponto você se oferece para ensinar o truque às crianças. Ainda temos de ver se a oferta é recusada! Com as crianças distribuídas em duplas, distribui-se cartas necessárias para um quadro 6x6.

Após formarem um quadrado 5x5 com suas cartas, pergunta-se:

Quantas cartas coloridas estão em cada linha e coluna?

Trata-se de um número par ou ímpar?

Verifique se as crianças se lembram do conceito de números ímpares e pares, e se eles reconhecem que zero é um número par.

Em seguida, adicione uma quinta carta a cada linha e coluna, certificando-se de que o número de cartas coloridas é par (se já é, então a carta extra deve ser branca, caso contrário, deve ser

colorida). O nome técnico para a carta extra é *carta de paridade*, e pode ser útil para ensinar as crianças o termo neste momento.

Agora mostre o que acontece se uma carta for virada - a linha e a coluna da carta virada terão um número ímpar de cartas coloridas, e assim a carta virada é aquela onde a linha e a coluna alteradas se cruzam. Cada membro da dupla pode agora se revezar em fazer o "truque".

Uma vez que eles estiverem confiantes com o truque, podem desejar para se reunir com outra dupla e fazerem um retângulo maior de cartas. Você pode até tentar juntar todas as cartas das crianças para fazer um quadro enorme.

Diga às crianças que as cartas de paridade são usadas para mostrar a ocorrência de um erro.

Variações e extensões

1. Em vez de cartas, você pode usar quase todo objeto que tenha dois "estados". Por exemplo, você pode usar moedas (caras ou coroas), paus (apontando para a esquerda ou para a direita) ou copos (para baixo ou para cima). Se a atividade estivesse relacionada à representação binária, as cartas poderiam ter um ZERO de um lado e UM do outro. Isso facilita a explicar o significado do exercício - que as cartas representam uma mensagem em binário, aos quais são adicionados bits de paridade para proteger a mensagem de erros.

2. Outro exercício interessante é considerar a carta inferior direita. Se você escolher para ser o correto para a coluna acima, então será correto para a linha à esquerda? (A resposta é sim, o que é oportuno, pois evita ter de se lembrar se o mesmo é referente à linha ou à coluna.) As crianças provavelmente podem "provar" isso para si mesmas, tentando encontrar um contra-exemplo.

3. A descrição do truque usa paridade par - exige um número par de cartões coloridos. Também é possível usar paridade ímpar, onde cada linha e coluna tem um número ímpar de cartões coloridos. No entanto, o cartão do lado direito somente funciona para a sua linha e coluna se os números de linhas e colunas são ambos pares ou ímpares. Por exemplo, um retângulo de 5x9 ou um retângulo de 12x4 funcionará bem, mas um retângulo de 3x4 não. As crianças podem ser convidadas a experimentar paridade ímpar e ver se eles podem descobrir o que está acontecendo com o canto inferior.

3.3 Um Truque com Números Binários e Matrizes

Nesta atividade o tema Códigos vem como uma curiosidade matemática, despertando o interesse por assuntos mais aprofundados e novos campos da matemática pura e aplicada. Ele pode ser aplicado a alunos do Ensino Médio e ser utilizado na introdução ao estudo de Matrizes (para gerar o interesse pelo conteúdo) ou finalizando essa sessão (na aplicação dos conteúdos estudados, como as propriedades e as operações com matrizes). Aqui nos baseamos em [3], [5], [10] e [13] para descrever as etapas do truque.

Aqui o truque dá-se com as 7 cartelas contendo números de 1 a 15, permitindo a mentira em uma das cartelas.

Conteúdos

Potências;

Sistemas de numeração;

Bases numéricas;

Base Binária;

Matrizes e suas Propriedades;

Divisibilidade.

Objetivos

1. Relembrar diferentes sistemas de numeração;
2. Aprofundar o estudo sobre Matrizes;
3. Aprofundar o estudo sobre a base binária;
4. Conhecer aplicações da base binária.

Duração

Três aulas.

Material necessário

Lápis;

Borracha;

Folhas de papel A4 (Cartões do anexo).

A mágica - Procedimentos Metodológicos

1ª Etapa

Pode-se iniciar a aula semelhantemente à direcionada ao Ensino Fundamental, porém agora, o professor apresenta aos alunos os cartões com numeração de 1 a 15 (cartões comuns) e pede a um dos alunos pense em um número e diga em quais cartões o número se encontra. Após descobrir o número deve haver um certo espanto pelos alunos. Continua com truque mais duas ou três rodadas, assim que tiver a total atenção dos alunos mostra os novos cartões e permite que o voluntário minta ou não uma vez. E agora para o maior espanto dos alunos ele continua acertando.

2ª Etapa

Como nesse momento os alunos devem estar ansiosos para saber como ele advinha, o professor explica como o truque usando os seguintes procedimentos:

1º - Separe os alunos em grupos de 3 integrantes e mostre os cartões simples numerados de 1 a 15 aos grupos. Explique como o truque inicial funcionou para a adivinhação.

2º - Novamente, peça para que um dos alunos escolha um número entre 1 e 15;

3º - Solicite que ele separe os cartões em que o valor escolhido está presente e some o primeiro valor do canto superior esquerdo de cada um dos cartões que ele escolheu inicialmente.

Após entenderem o truque, pergunte:

Questão para os alunos: O truque com esses cartões funcionaria se o voluntário mentisse? Dê um exemplo com mentira utilizando os cartões simples.

3ª Etapa

Neste momento os grupos devem discutir e tentar explicar por que esse procedimento funciona. Esperamos que os alunos notem que a primeira célula de cada cartela é uma potência de 2. Os outros valores que compõem cada cartela são aqueles que apresentam, em sua representação binária, a mesma potência de 2 que aparece na sua primeira célula.

Sugira que cada grupo escreva uma tabela com as somas realizadas durante o truque para buscar as informações que desvendem o mistério. Caso ache necessário, peça também para escreverem as parcelas na sua forma fatorada, conforme o exemplo:

Os grupos devem formular hipóteses sobre a explicação da mágica. Essas hipóteses podem ser

Número Escolhido	Soma dos números iniciais	Soma das potências de base 2
15	8+4+2+1	$2^3 + 2^2 + 2^1 + 2^0$
9	8+1	$2^3 + 2^0$

Tabela 3.2: Exemplos

apresentadas para a turma e uma discussão decide as que são válidas e as que não são. Por fim, introduza a notação da solução da mágica com potências 2 caso os grupos ainda não o tenham feito e mostre que "todo número natural pode ser escrito como a soma de diversas potências distintas de 2" (não há necessidade de provar esse resultado).

4ª Etapa

Agora peça aos alunos que montem uma tabela contendo todos os números de 1 a 15 e sua representação por meio da soma de diversas potências 2, conforme o exemplo anterior.

Aqui temos a tabela pronta.

Número em Base 10	Somas	Soma de Potências de 2
1	1	2^0
2	2	2^1
3	2+1	$2^1 + 2^0$
4	4	2^2
5	4+1	$2^2 + 2^0$
6	4+2	$2^2 + 2^1$
7	4+2+1	$2^2 + 2^1 + 2^0$
8	8	2^3
9	8+1	$2^3 + 2^0$
10	8+2	$2^3 + 2^1$
11	8+2+1	$2^3 + 2^1 + 2^0$
12	8+4	$2^3 + 2^2$
13	8+4+1	$2^3 + 2^2 + 2^0$
14	8+4+2	$2^3 + 2^2 + 2^1$
15	8+4+2+1	$2^3 + 2^2 + 2^1 + 2^0$

Tabela 3.3: Somas de potências de bases 2.

Com a tabela construída, fale sobre a representação binária dos números decimais.

Comente com a turma sobre nosso sistema de numeração, o decimal, que utiliza apenas 10 algarismos para representar qualquer valor inteiro positivo e, na escrita desses valores, a posição que cada algarismos ocupa no número altera seu valor.

De maneira geral temos: $a, b \in \mathbb{N}$, com $b > 1$, existem números naturais $r_0, r_1, r_2, \cdots, r_n < b$, univocamente determinados, tais que

$$a = r_n b^n + r_{n-1} b^{n-1} + \cdots + r_2 b^2 + r_1 b^1 + r_0 b^0$$

Da mesma forma o sistema binário também é posicional, mostre exemplos de representação binária e que nesse sistema os algarismos são apenas zeros e uns. Por exemplo:

$$1011 \quad = \quad 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \quad = \quad 8 + 0 + 2 + 1 \quad = \quad 11$$

este valor é 11 na base dez.

Mencione que para converter um número da base decimal para a base binária, basta realizar divisões sucessivas por 2; os restos dessa divisão fornecerão os algarismos do número na nova base.

Exemplo 3.2. Vamos converter 75_{10} para base binária:

$$\begin{aligned} 75/2 &= 37, \textit{ resto } 1 \\ 37/2 &= 18, \textit{ resto } 1 \\ 18/2 &= 9, \textit{ resto } 0 \\ 9/2 &= 4, \textit{ resto } 1 \\ 4/2 &= 2, \textit{ resto } 0 \\ 2/2 &= 1, \textit{ resto } 0 \\ 1/2 &= 0, \textit{ resto } 1 \end{aligned}$$

Portanto, $75_{10} = 1001011_2$

Agora que já escreveram números binários acrescentar o valor do binário a tabela anterior, assim temos:

Assim que terminar a tabela mostre como foram criados os cartões do jogo simples (sem a mentira).

Esse ponto será a introdução do truque com a mentira.

5ª Etapa

Agora é hora de falar sobre códigos corretores de erros, codificação e decodificação.

O que pretende-se aqui nessa etapa é que o aluno familiarize-se com o tema de códigos e compreenda sua utilização.

Assim o professor pode fazer uso do Capítulo 2 deste trabalho, falando assim sobre códigos corretores de erros e mostrar as matrizes.

Número em Base 10	Somas	Soma de Potências de 2	Base 2	Matriz de Código Binário
1	1	2^0	1	$[\ 0\ \ 0\ \ 0\ \ 1\ ]$
2	2	2^1	10	$[\ 0\ \ 0\ \ 1\ \ 0\ ]$
3	2+1	$2^1 + 2^0$	11	$[\ 0\ \ 0\ \ 1\ \ 1\ ]$
4	4	2^2	100	$[\ 0\ \ 1\ \ 0\ \ 0\ ]$
5	4+1	$2^2 + 2^0$	101	$[\ 0\ \ 1\ \ 0\ \ 1\ ]$
6	4+2	$2^2 + 2^1$	110	$[\ 0\ \ 1\ \ 1\ \ 0\ ]$
7	4+2+1	$2^2 + 2^1 + 2^0$	111	$[\ 0\ \ 1\ \ 1\ \ 1\ ]$
8	8	2^3	1000	$[\ 1\ \ 0\ \ 0\ \ 0\ ]$
9	8+1	$2^3 + 2^0$	1001	$[\ 1\ \ 0\ \ 0\ \ 1\ ]$
10	8+2	$2^3 + 2^1$	1010	$[\ 1\ \ 0\ \ 1\ \ 0\ ]$
11	8+2+1	$2^3 + 2^1 + 2^0$	1011	$[\ 1\ \ 0\ \ 1\ \ 1\ ]$
12	8+4	$2^3 + 2^2$	1100	$[\ 1\ \ 1\ \ 0\ \ 0\ ]$
13	8+4+1	$2^3 + 2^2 + 2^0$	1101	$[\ 1\ \ 1\ \ 0\ \ 1\ ]$
14	8+4+2	$2^3 + 2^2 + 2^1$	1110	$[\ 1\ \ 1\ \ 1\ \ 0\ ]$
15	8+4+2+1	$2^3 + 2^2 + 2^1 + 2^0$	1111	$[\ 1\ \ 1\ \ 1\ \ 1\ ]$

Tabela 3.4: Conversões

$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad e \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Nessa etapa o professor pode pedir aos alunos que calculem o produto da matriz geradora G por cada "palavra a ser enviada" (os números de 1 a 15 em binários).

Após o cálculo pede que troquem cada 0 por N (não) e 1 por S (sim). Após esse procedimento serão confeccionados os cartões.

Fechamento

Finaliza-se essa atividade seguindo os passos descritos na Sessão 2.4.1 que mostram a confecção dos cartões com abas contextualizando o tema "Códigos Binários e truques de mágica" com o conteúdo de sala de aula (Matrizes). Após as construções convidamos os alunos a fazerem o jogo com outros alunos de outras séries, testando o novo truque que aprenderam.

Referências Bibliográficas

[1] BELL, T.; WITTEN, I. H.; FELLOWS M. *Computer Science Unplugged ... off-line activities and games for all ages.* Disponível em: <http://csunplugged.org/wp-content/uploads/2015/01/unplugged-book-v1.pdf> Acesso em: 19 março 2017.

[2] BRASIL, *Secretária de Educação Fundamental, Parâmetros Curriculares Nacionais*, 3 ed. Brasília: MEC, vol 1, 1997.

[3] EHRENBORG, R., *Decoding the Hamming code*, Math Horizons, special issue on Codes, Cryptography and National Security, University of Kentucky, Lexington, 2006, 16-17.

[4] FERREIRA, S. *Sistema binário - Parte I.* Disponível em: <http://www.linhadecodigo.com.br/artigo/1648/sistema-binario-parte-i.aspx> Acesso em: 03 outubro 2016.

[5] GUIMARÃES, R. S. *Mágica das Cartelas.* Disponível em: <http://m3.ime.unicamp.br/recursos/1019> Acesso em: 05 setembro 2016.

[6] HAMMING, R. W., *Interview*, February 3 - 4, 1977.

[7] HEFEZ, A. *Aritmética.* Rio de Janeiro: SBM, 2014. 330p. (Coleção PROFMAT).

[8] HEFEZ, A.; VILLELA, M. L. T. *Códigos Corretores de Erros.* 2.ed. Rio de Janeiro: IMPA, 2008. 216p

[9] LAVOR, C. C. ; ALVES, M. M. S. ; SIQUEIRA, R. M. ; COSTA, S. I. R. , *Uma Introdução à Teoria de Códigos*, São Carlos, SP : SBMAC, 2006.

[10] MATEER, T., *A magic trick based on the Hamming Code*, *Math Horizons 21* (Novembro, 2013), 9-11. Material suplementar em: <https://mthsc.clemson.edu/misc/MAM_2014/mh-09-11-Mateer.pdf>.

[11] MILES, C. P. *Breve introdção à Teoria dos Códigos Corretores de Erros*. Disponível em: <http://www.sbm.org.br/docs/coloquios/NE-1.04.pdf> Acesso em: 05 abril 2017.

[12] MIRANDA, D. S., *Códigos corretores de erros e empacotamentos de discos*. Trabalho de Conclusão(Mestrado Profissional), Departamento de Matemática da Universidade Federal Rural de Pernambuco, Recife-PE, 2013.

[13] MOREIRA, E. M., PICADO, J., *Truques e Magia com Códigos Algébricos*, Universidade de Coimbra, Porto: Gazeta de Matemática, n. 0175, 2015

[14] POOLE, D. *Álgebra linear*. São Paulo: Pioneira Thomson Learning, 2004.

[15] SILVEIRA, R. B. R., *Códigos corretores de erros: exemplos da matemática aplicada em situações do cotidiano*, Dissertação (mestrado) - Universidade Federal Rural do Rio de Janeiro, Curso de Mestrado Profissional em Matemática em Rede Nacional, Rio de Janeiro, 2015.

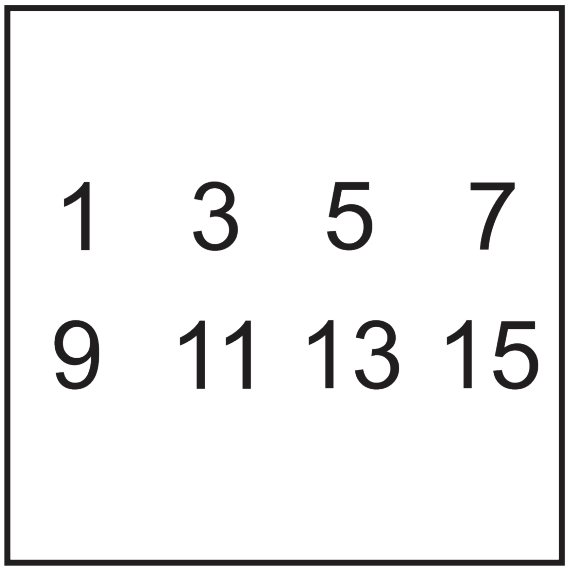
[16] SLINKO, A. M. *Algebra for Applications: Cryptography, Secret Sharing, Error-Correcting, Fingerprinting, Compression*, Department of Mathematics The University of Auckland New Zealand: Springer, 2015.

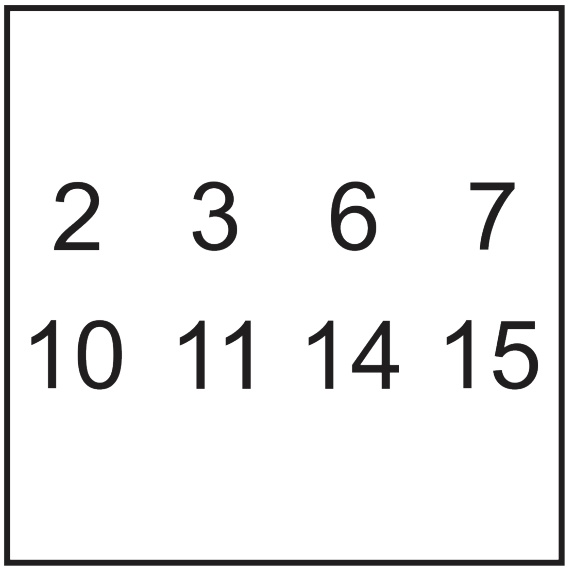
[17] SODRÉ, U. *Álgebra: Classes modulares*, 2006. Disponível em: <http://www.uel.br/projetos/matessencial/superior/algebra/modular.htm>. Acesso em: 03 outubro 2016.

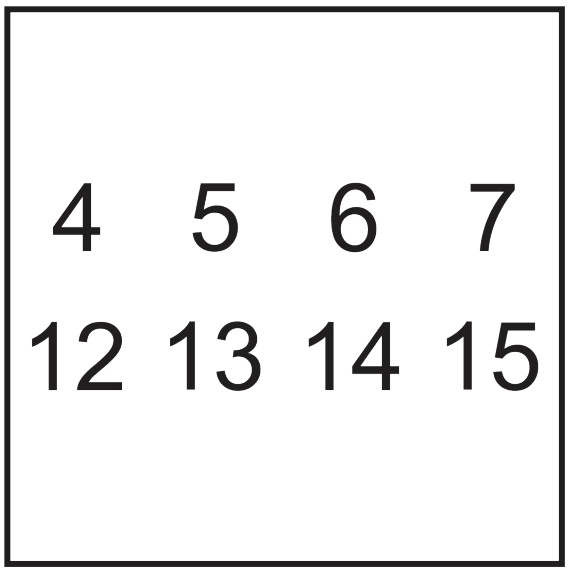
[18] VANSTONE, S. A.; OORSHOT, P. C. V., *An introduction to error correcting codes with applications*. Norwell, Massachusetts: Kluwer Academic Publishers, 1989. 1-16

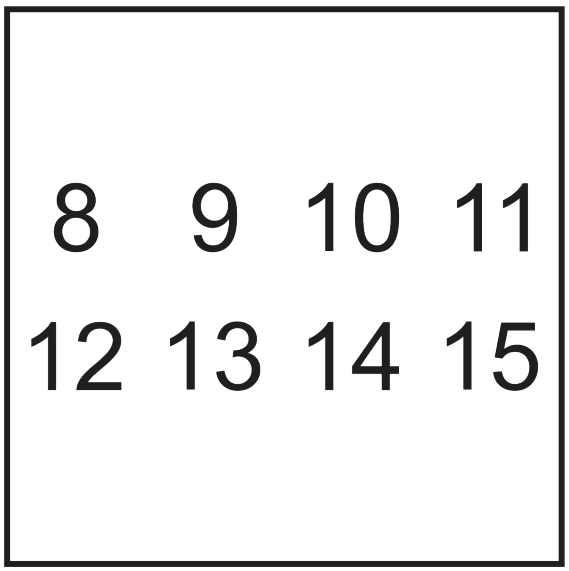
Anexos

Cartões do Truque de Mágica com números de 1 a 15.









Cartões do Truque de Mágica com números de 1 a 31.

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

Cartões do Truque de Mágica com números de 1 a 63.

1	3	5	7	9	11
13	15	17	19	21	23
25	27	29	31	33	35
37	39	41	43	45	47
49	51	53	55	57	59
61	63				

8	9	10	11	12	13
14	15	24	25	26	27
28	29	30	31	40	41
42	43	44	45	46	47
56	57	58	59	60	61
62	63				

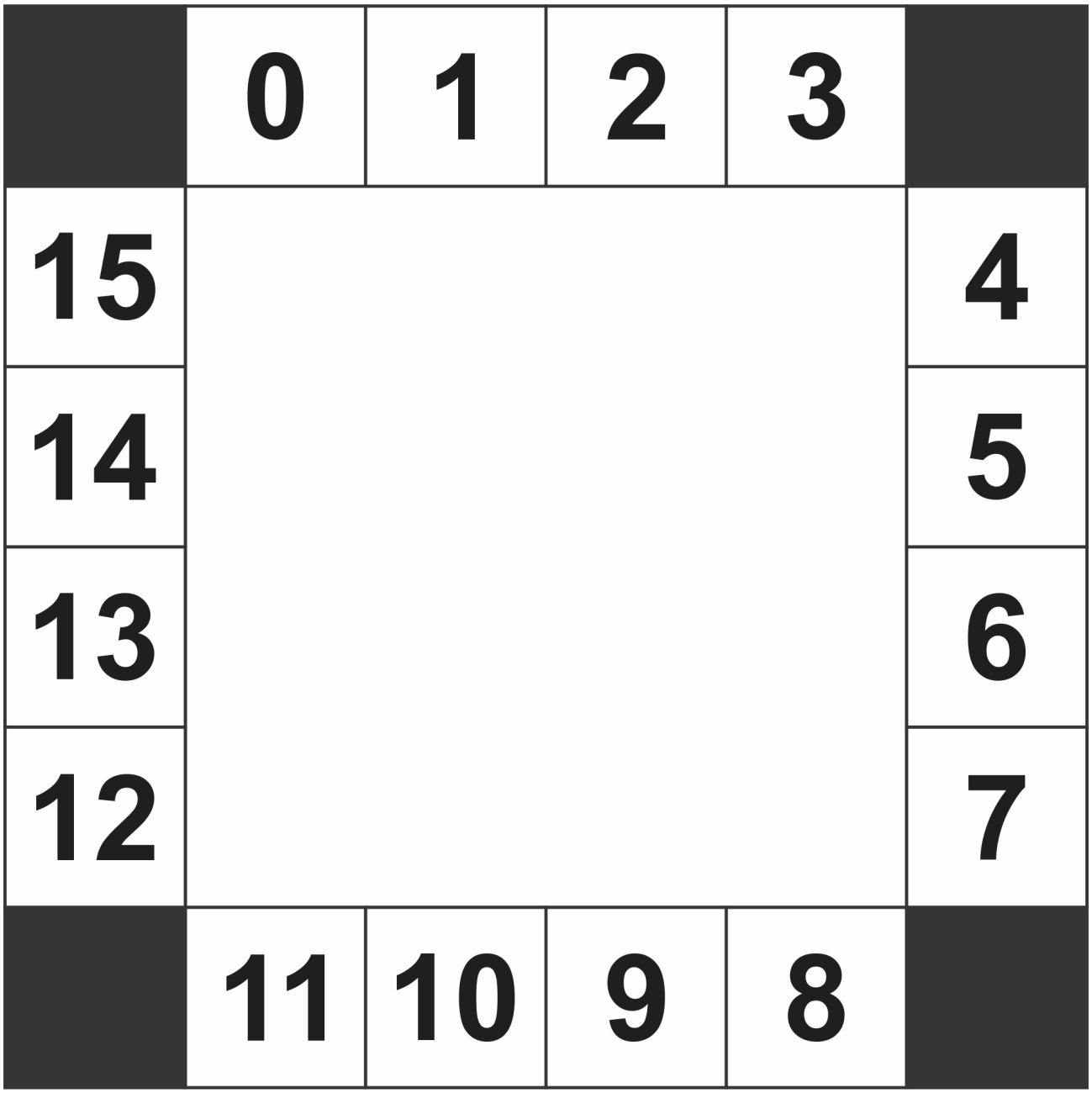
2	3	6	7	10	11
14	15	18	19	22	23
26	27	30	31	34	35
38	39	42	43	46	47
50	51	54	55	58	59
62	63				

16	17	18	19	20	21
22	23	24	25	26	27
28	29	30	31	48	49
50	51	52	53	54	55
56	57	58	59	60	61
62	63				

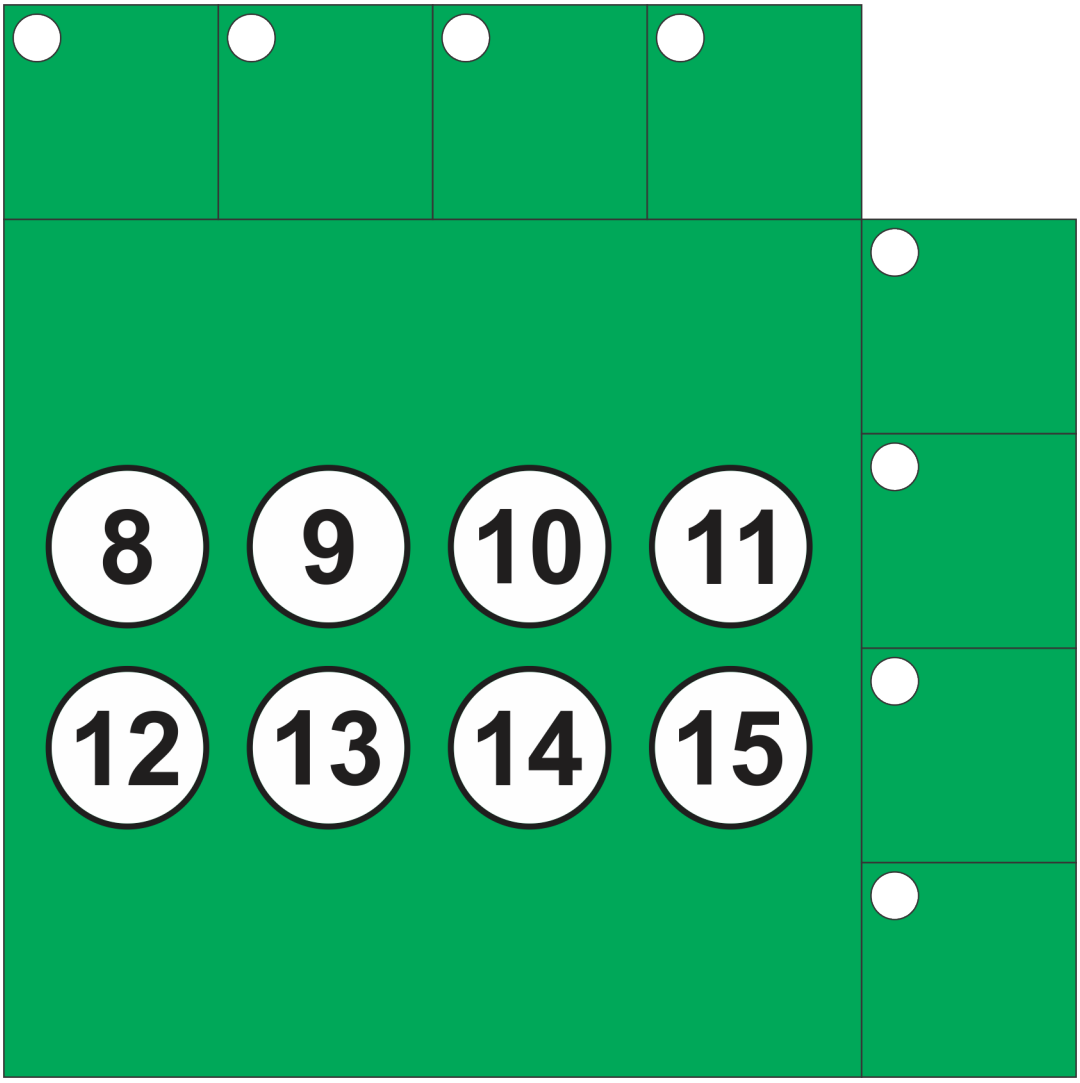
4	5	6	7	12	13
14	15	20	21	22	23
28	29	30	31	36	37
38	39	44	45	46	47
52	53	54	55	60	61
62	63				

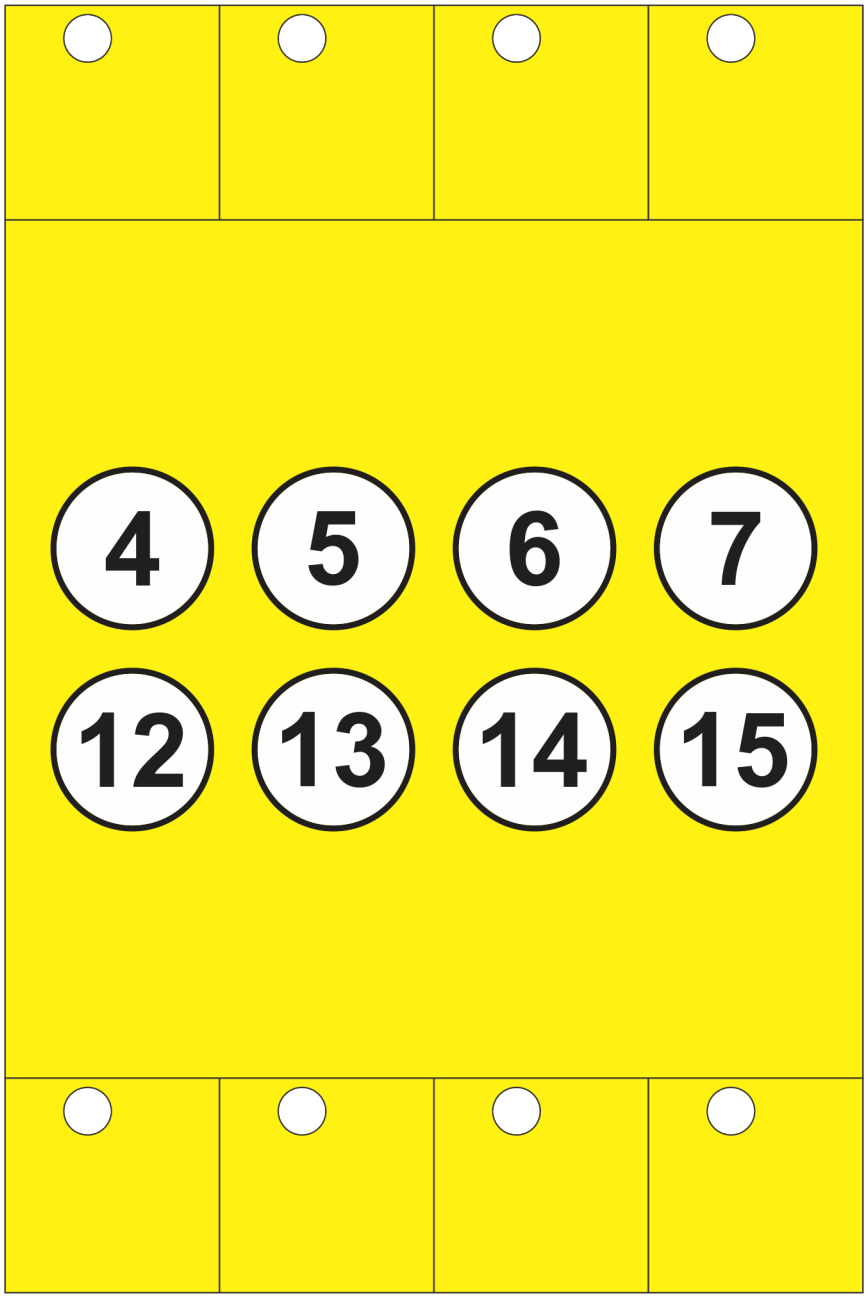
32	33	34	35	36	37
38	39	40	41	42	43
44	45	46	47	48	49
50	51	52	53	54	55
56	57	58	59	60	61
62	63				

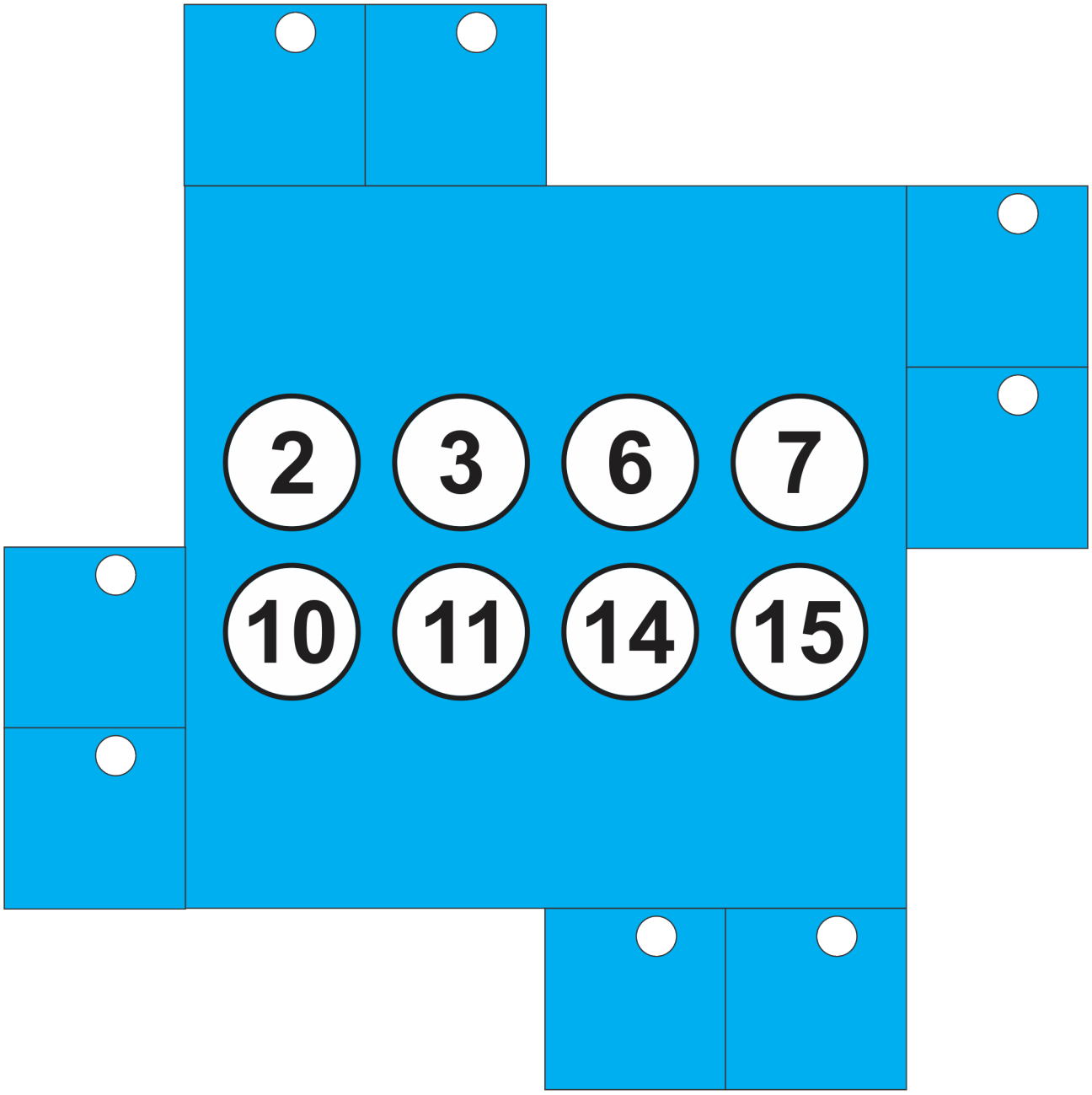
Cartões do Truque de Mágica que permitem a mentira.

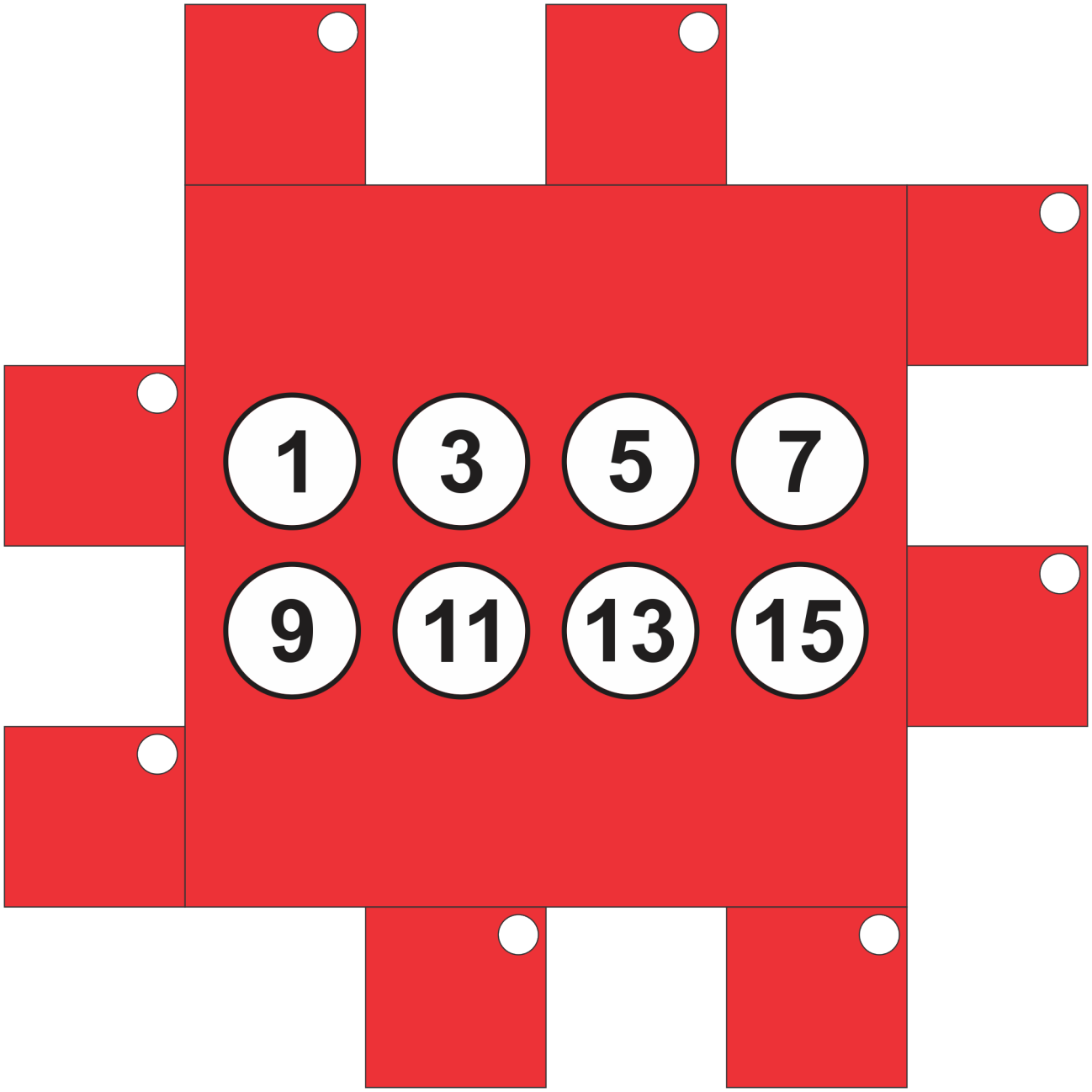


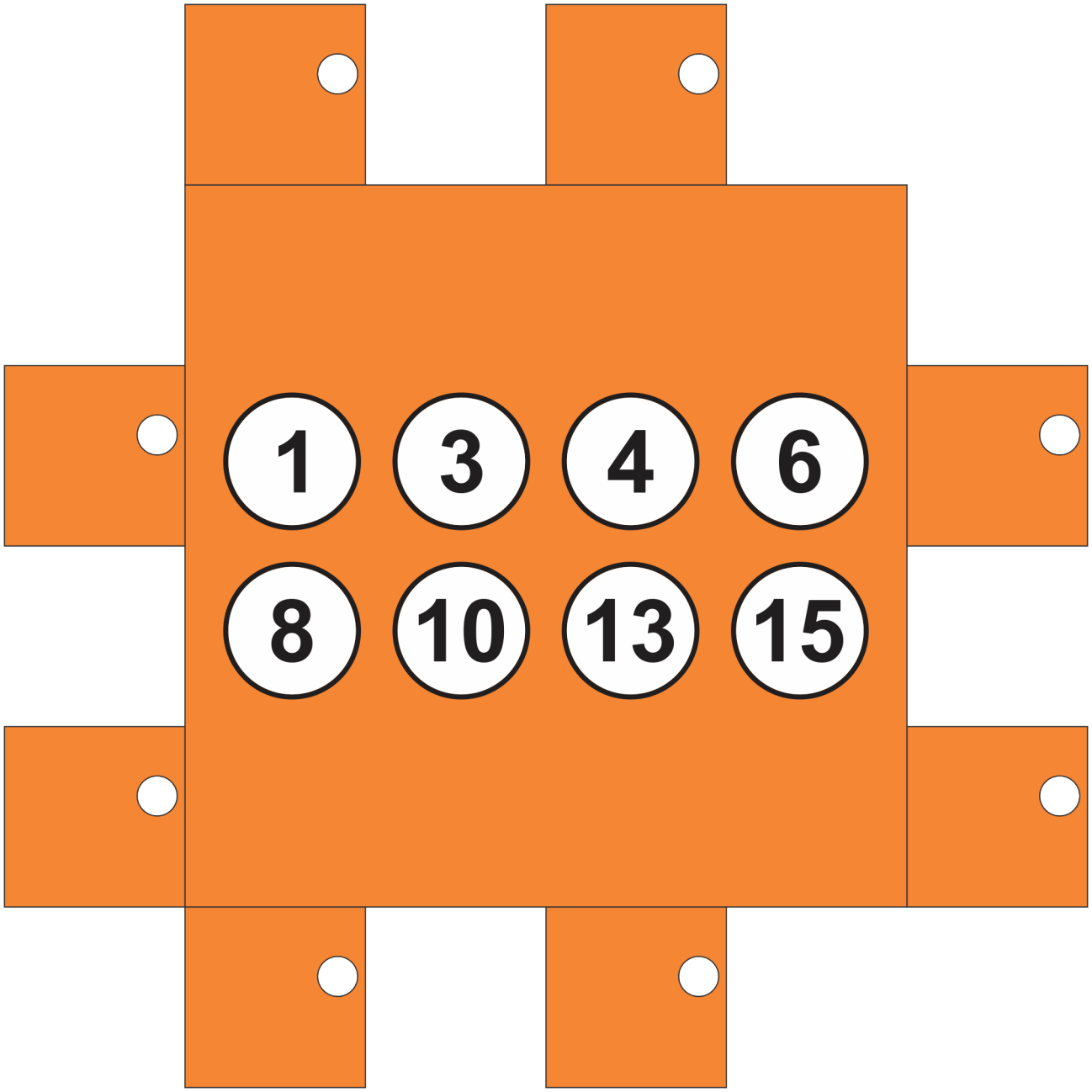
Cartão-base

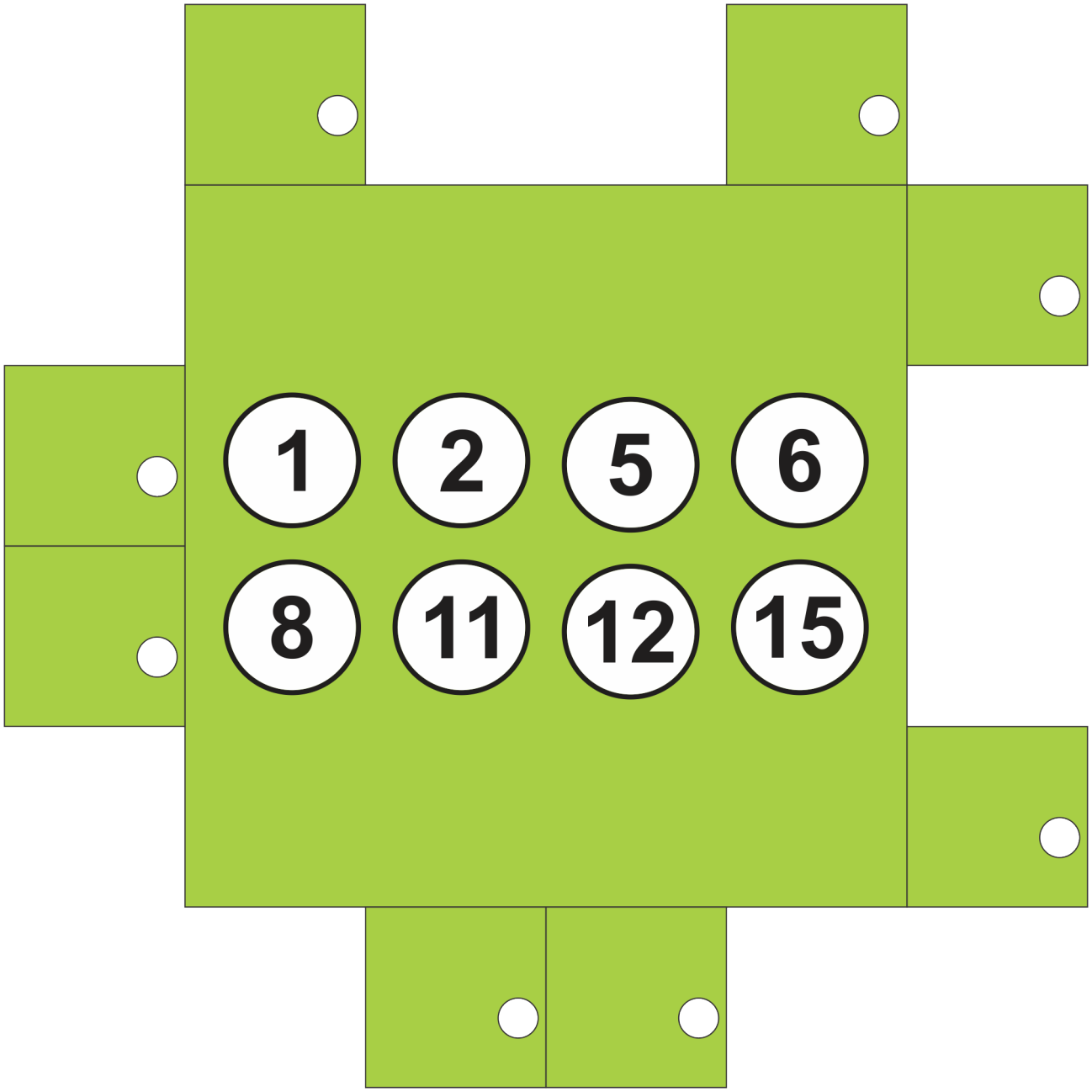

8
9
10
11
12
13
14
15

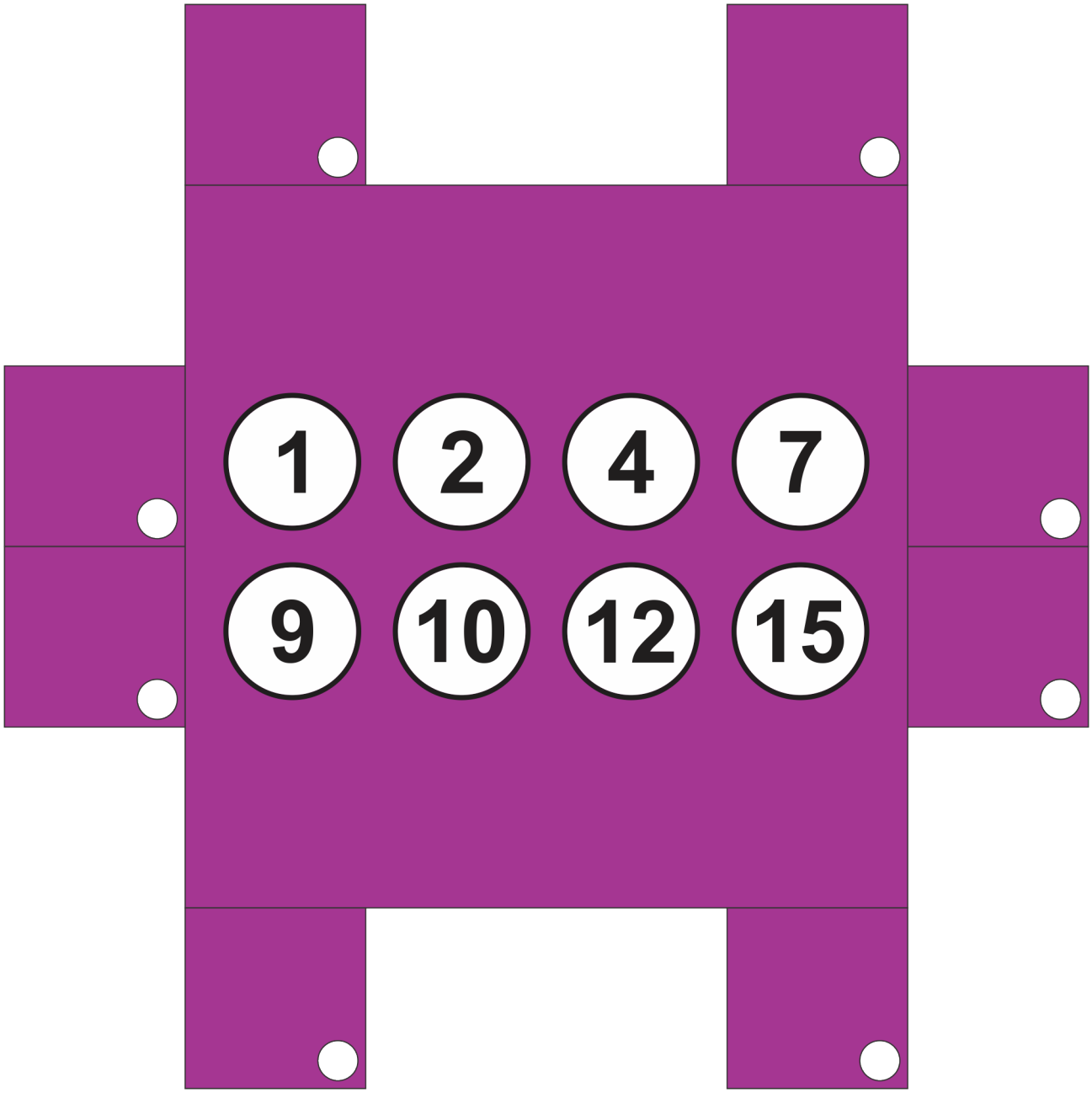

4
5
6
7
12
13
14
15


2
3
6
7
10
11
14
15


1
3
5
7
9
11
13
15


1
3
4
6
8
10
13
15


1
2
5
6
8
11
12
15


1
2
4
7
9
10
12
15