Introdução à Ciência da Computação

Módulo 1

Luis Retondaro

CEFET-RJ
Campus Petrópolis

August 15, 2023

Sumário

1 Visão geral

2 Representação

3 Algoritmos

Visão geral

“A história não é exclusivamente caos ou acaso: existe no comportamento humano um certo grau de ordem e padrão observáveis de uma regularidade parcialmente previsível. ”

The Social Sciences in Historical Study (vários autores)

Objetivo do curso

Ao final do curso (disciplina) você será capaz de:

- Compreender a área e a atuação da computação como atividade meio e fim.
- Reconhecer e operar representações de dados típicas
- Aumentar o poder de abstração para soluções de problemas
- Formular algoritmos para resolver diversos problemas com o computador.
- Analisar, refinar, organizar, otimizar e depurar algoritmos básicos.
- Programar e executar algoritmos fundamentais no computador escritos na linguagem Python.

Visão geral

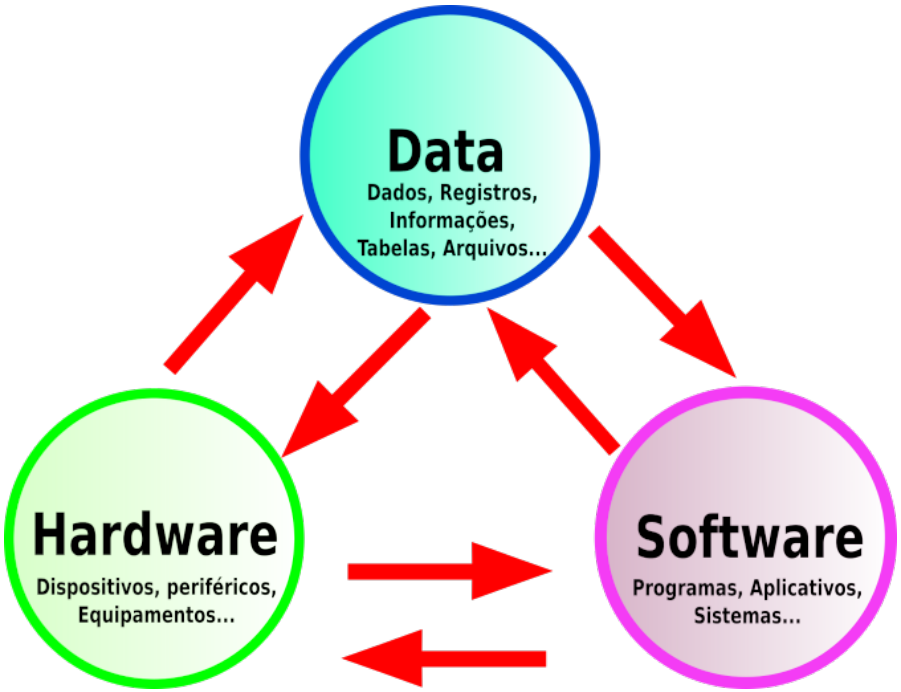


Figure: Principais elementos da computação.

Computação

O que é um computador?

- Dispositivo (eletrônico) usado para a solução de problemas.
- Manipulação, armazenamento e acesso a dados.
- Circuitos digitais binários (0 e 1) - bits.
- Macro-blocos principais:
  - Memória, unidade central de processamento e módulo de entrada e saída.

Arquitetura de computadores

Arquitetura de John Von Neumann

Figure: Simplificação da arquitetura de Von Neumann.

John Von Neumann - matemático húngaro (1903-1957) - um dos mais importantes do século XX.

Arquitetura Von Neumann

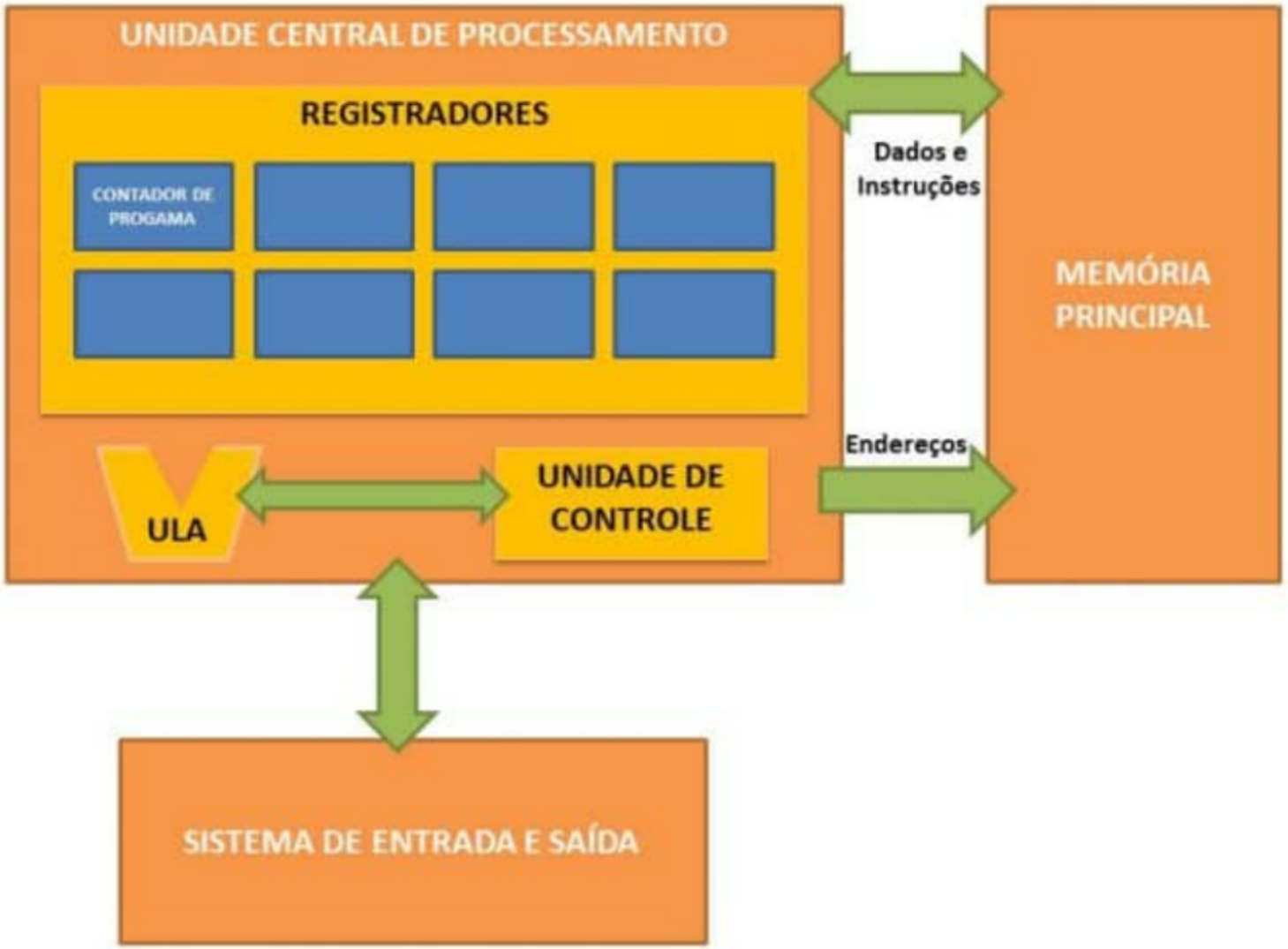


Figure: Alguns detalhes da arquitetura de Von Neumann.

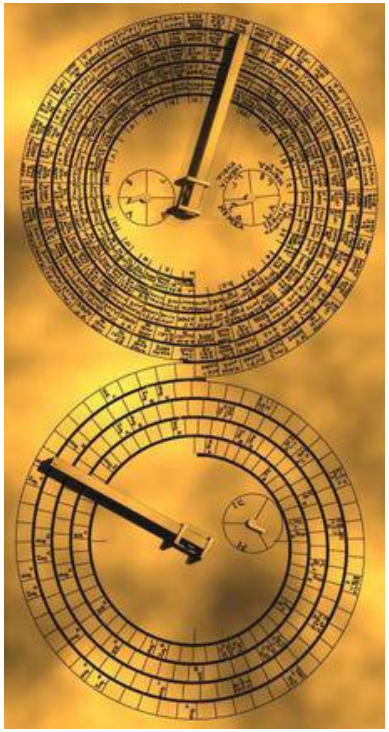
Um pouco de história da computação

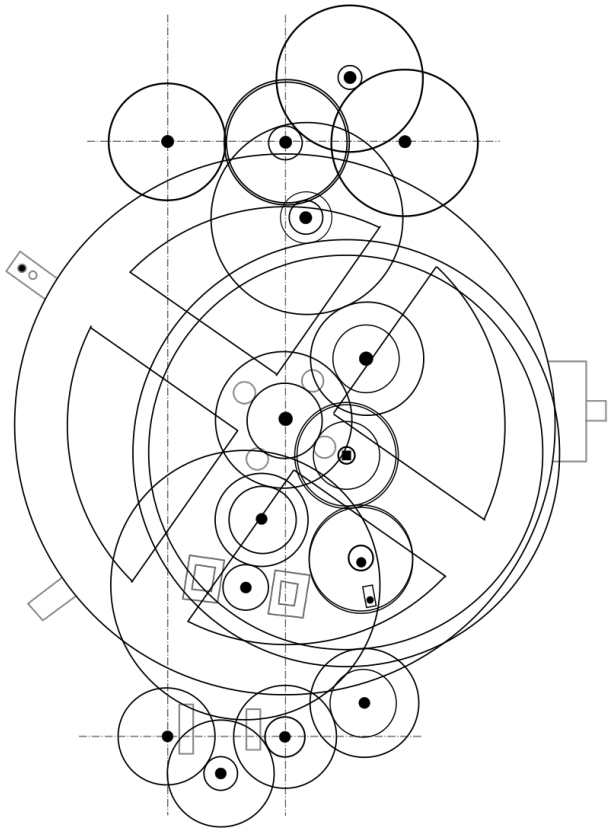
- Muito antes do conceito do computador moderno, os seres humanos já tentavam criar dispositivos para ajudar em tarefas dispendiosas ou perigosas.

Exemplo: Ábaco.

Para cálculos simples.

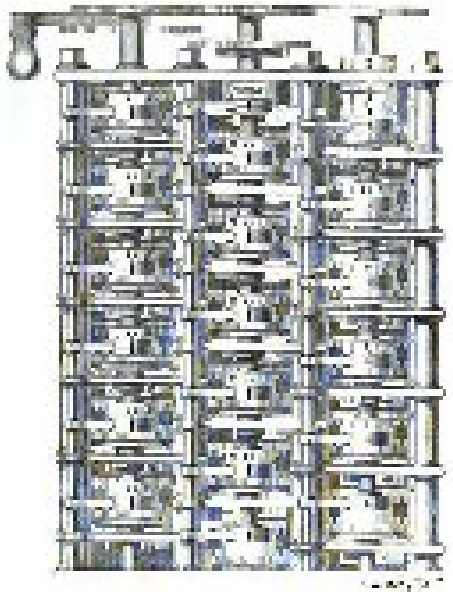
Máquina de Anticítera (≈200 a.C.)





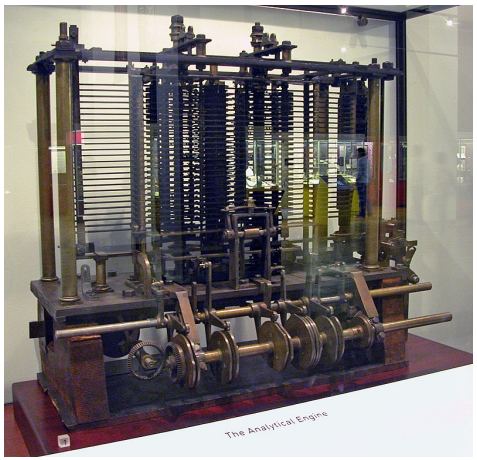
- Usada para prever posições astronômicas e eclipses;
- Progressões aritméticas;
- Mecanismo extremanente complexo com diversas engrenagens.

Máquina de Diferencial de Babbage (1822)



- Calcular funções polinomiais;
- Capaz de receber, processar, guardar e exibir dados.

Máquina de Analítica de Babbage (1822)



- Computador mecânico, com propósito de realizar cálculos genéricos.
- Incluia uma unidade lógica aritmética, fluxo de controle, memória.
- Sua construção nunca foi concluída.

Outros nomes da história

Blaise Pascal (1623-1662)

- Aos dezesseis anos, em 1640, publicou a obra Essai sur les coniques , formulando um dos teoremas básicos da Geometria Projetiva, conhecido, desde então, como o Teorema de Pascal.
- Aos dezenove anos, com o objetivo de auxiliar seu pai, projetou a primeira calculadora, patenteada como o nome de “La Pascaline”
- O primeiro a cogitar da utilização da linguagem binária como base da programação de cálculos e processamento de operações lógicas, fazendo-o um dos grandes precursores da Inteligência Artificial.
- Aos vinte e quatro anos, em 1648, Pascal voltou-se para a física, pesquisando a mecânica dos fluídos e esclarecendo os conceitos de pressão e vácuo.

Outros nomes da história

Gottfried Wilhelm Leibniz (1646-1716)

- Foi um proeminente polímata e filósofo alemão e figura central na história da matemática
- Concebeu as ideias de cálculo diferencial e integral, independente de Isaac Newton
- Refinou o sistema de números binários que se tornaria a base de todos os computadores digitais

Outros nomes da história

Herman Hollerith (1860-1929)

- Inventou a máquina que fazia a leitura de cartões de papel perfurados em código BCD (Binary Coded Decimal)
- Em 1911, quatro corporações, incluindo a firma de Hollerith, se fundiram para formar a Computing Tabulating Recording Corporation. Sob a presidência de Thomas J. Watson, ela foi renomeada para IBM.

Outros nomes da história

Alan Turing (1912-1952)

- Desenvolveu a moderna ciência da computação teórica, proporcionando uma formalização dos conceitos de algoritmo e computação com a máquina de Turing, que pode ser considerada um modelo de um computador de uso geral.
- Pai da ciência da computação teórica e da inteligência artificial.
- Após a segunda guerra Turing trabalhou no Laboratório Nacional de Física, onde projetou o Mecanismo de Computação Automática, um dos primeiros projetos para um computador de programa armazenado

Sugestão interessante

Doxiadis A., Papadimitriou C. **"Logicomix"**, 2010. WMF Martins Fontes.

1a Geração – Computadores a válvula (1941–1957)

- Destinados apenas a funções de cálculos
- Usados na resolução de problemas específicos (cada máquina possuia o seu próprio código; novas funções requeriam a reprogramação completa do computador)
- Eram gigantescos e sofriam constantemente com o superaquecimento
- Cada válvula acesa ou apagada representava uma instrução à máquina
- Exemplo: ENIAC (Electronic Numerical Integrator and Computer), criado em 1945; considerado por muitos o primeiro computador eletrônico digital

2a Geração – Computadores a transistores (1958-1964)

- Transistores: componentes eletrônicos criados em 1947, pela Bell Laboratories, a partir de silício (material utilizado até hoje em placas e outros componentes)
- Vantagens dos transistores sobre as válvulas: são menores, dissipam menos calor e não se quebram por choque físico
- Computadores 100 vezes menores que os da 1a geração
- Mais econômicos (em consumo de energia e preço das peças)
- Programáveis em linguagem de montagem (Assembly), em substituição à linguagem de máquina
- Exemplo: IBM 7094 (1962)

3a Geração – Circuitos Integrados (1965–1970)

- Circuitos integrados feitos de silício permitiram o miniaturização de componentes eletrônicos
- Aumentos significativos na velocidade e eficiência dos computadores
- Teclados e monitores para a entrada e saída de dados
- Primeiros sistemas operacionais (não gráficos)
- Capacidade de upgrade nas máquinas
- Exemplo: IBM 360 (1965)

4a Geração – Microprocessadores (1971–1981)

- Microprocessador = Toda CPU dentro de um circuito integrado
- Compactação do computador = PC e afins
- Uso do *floppy disk* como unidade de armazenamento
- Surgimento de diversas linguagens de programação de todos os tipos
- Redes de transmissão de dados para a interligação de computadores
- Tecnologia *LSI (Large Scale Integration)*
- Exemplo: Sinclair Z81 (1981)

5a Geração – VLSI e ULSI (1981–?)

- Componentes com altíssima escala de integração
- Inteligência Artificial
- Processamento de Linguagem natural
- Teleprocessamento, internet
- Realidade virtual
- etc
- Exemplo: supercomputador Santos Dumont (2019)

Alguns outros modelos históricos

Figure: XT-8088 (1983)

Figure: Apple Lisa (1984)

Alguns outros modelos históricos

Figure: AT-80286 (1984)

Figure: TK 90x (1985)

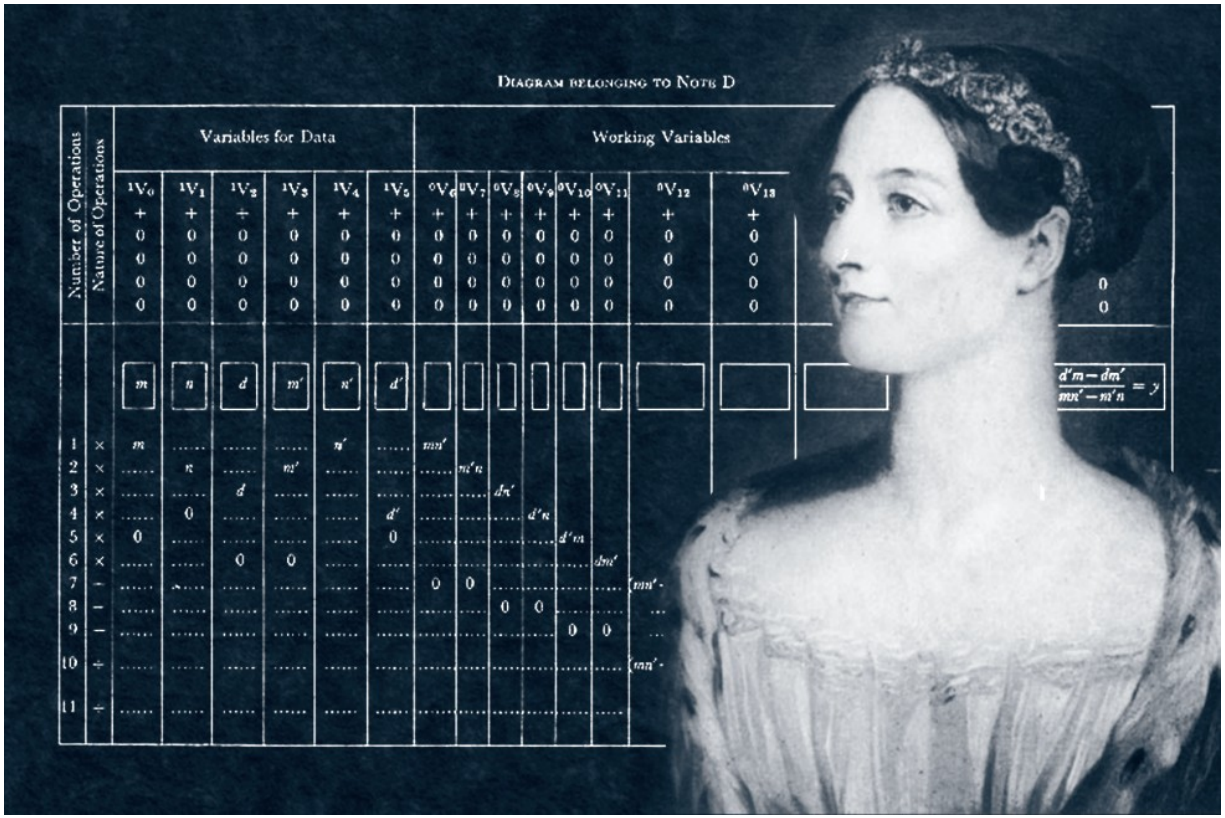
O que é um programa de computador?

> Um programa de computador que contém estruturas dinâmicas de acesso e armazenamento de dados que manipulam a máquina para processá-los e fornecer resultados formatados e solicitados por uma determinada demanda, seja genérica ou específica, de uma comunidade de usuários.

- Se a definição anterior for aceita, o primeiro software surgiu na Inglaterra em 1948, baseado num sistema criado por John von Neumann.
- Entretanto...

A primeira programadora

- 100 anos antes de Neumann, esse conceito de software já havia sido imaginado, na teoria, pela Condessa Ada Lovelace - matemática (1815-1852).
- Por isso, muitos estudiosos consideram-na a primeira programadora da história.
- Escreveu um algoritmo compatível com a máquina de Babbage para que essa fosse capaz de calcular os Números de Bernoulli (1842).



O Primeiro computador brasileiro e outras iniciativas

projeto Lourinha - IME - (1958)

Zezinho - ITA - (1961) *(O primeiro computador)

Patinho Feio - USP - (1972)

Cobra computadores - a primeira empresa (1974)

Cobra 530 - o primeiro industrializado 100% nacional

Comece a pesquisa aqui: Lourinha, Patinho Feio e Cobra

Representação

“Na ciência e em todo trabalho de criação nós falhamos repetidas vezes. Normalmente para cada ideia bem sucedida há dúzias de outras que não funcionaram.”

John Backus, criador do Fortran e da programação funcional, em seu discurso ao receber o prêmio Charles Stark Draper, 1994.

Organização

O computador organiza para acessar e otimizar execução de rotinas

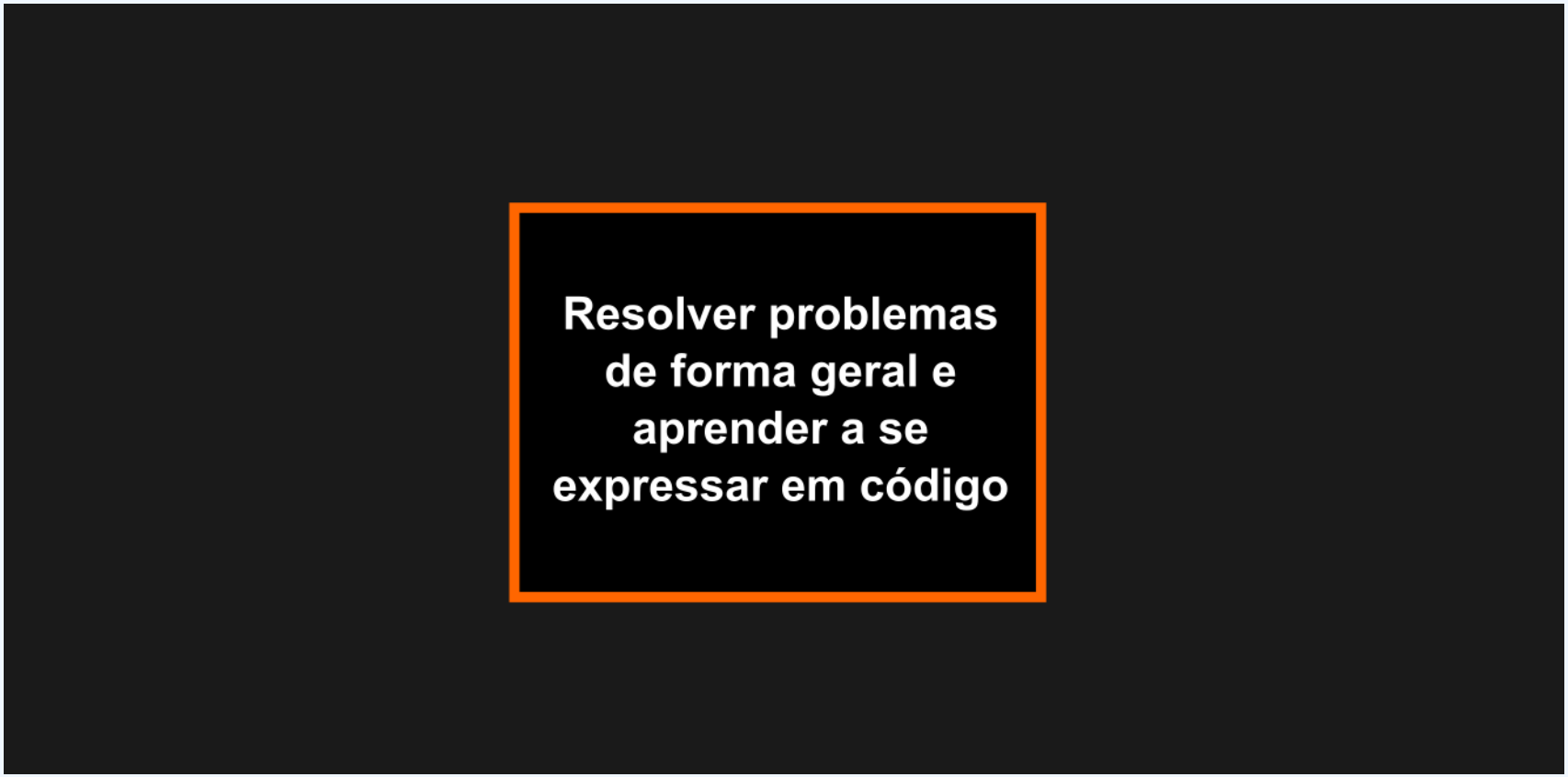
- Memória (bytes)
- Estruturas de dados (links, listas...)
- Instruções (códigos, comandos...)
- Dados (tipos, formatos)

Ciência da Computação

Problema

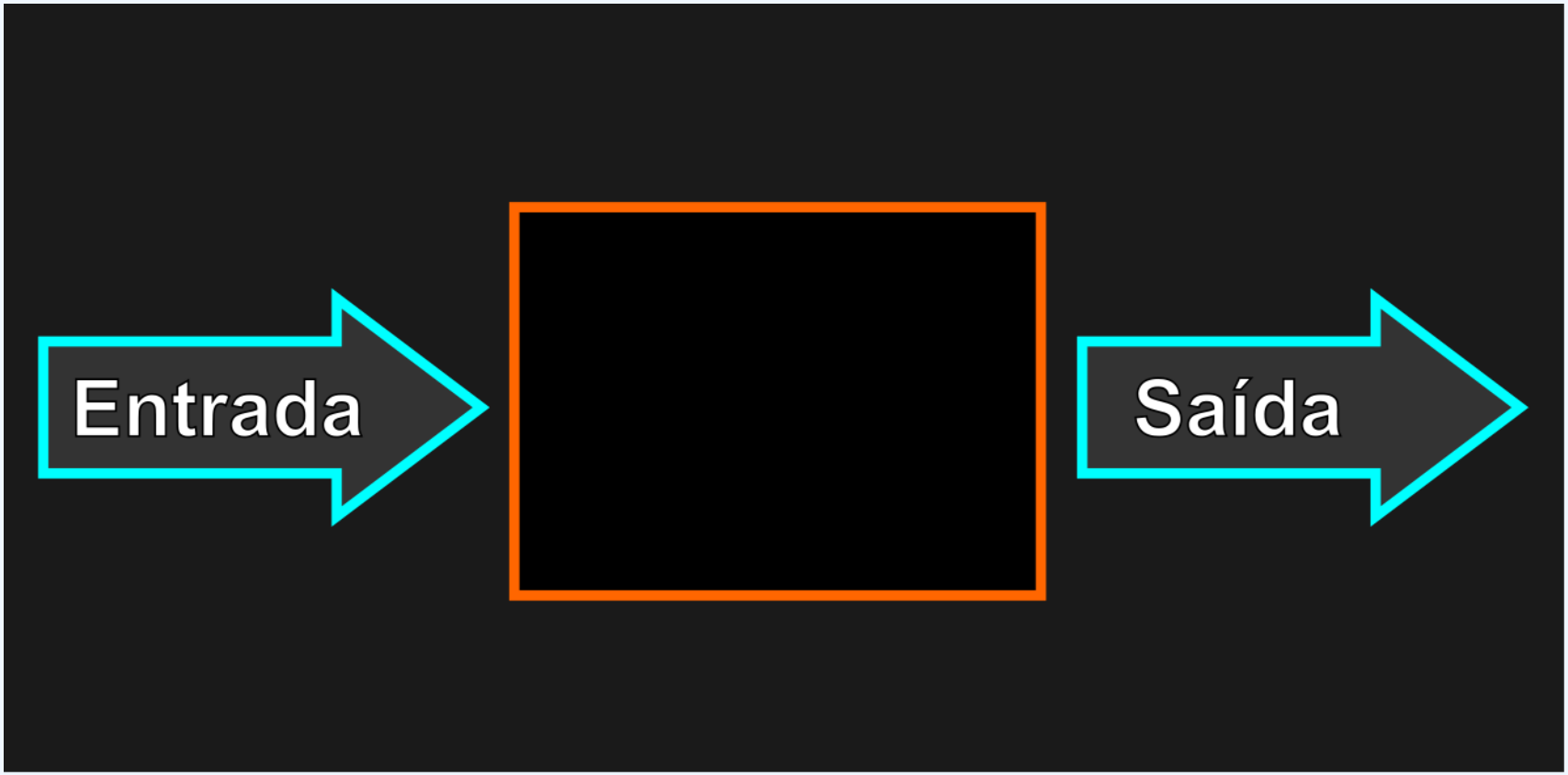
Ciência da Computação

Propósito



Ciência da Computação

A solução de Problemas



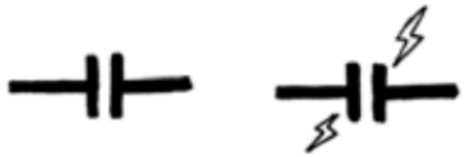
Sistema Unário - Base-1

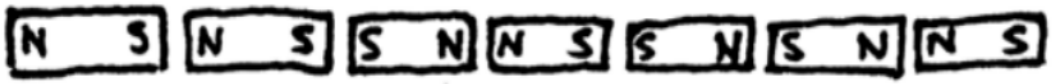
Simples e eficaz

- Tipicamente para contagem
- Ex.: Uso dos dedos das mãos (dígitos)
- 1,2,3,4,5...

Sistema Binário - Base-2

Fundamentação

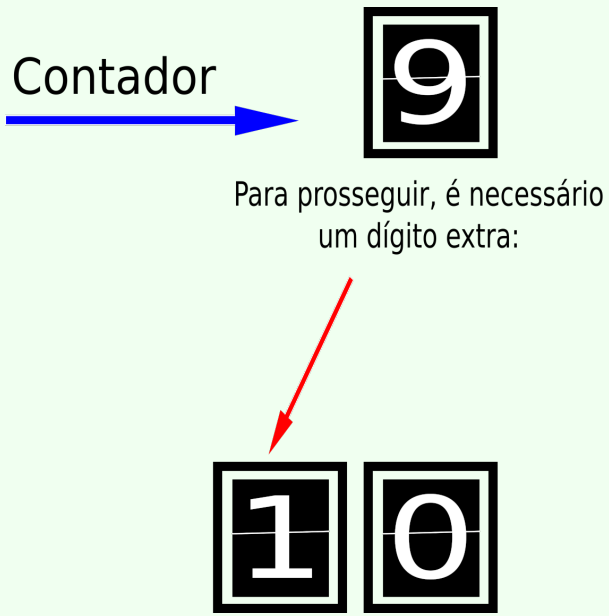




Sistemas de numeração

Sistema de base decimal

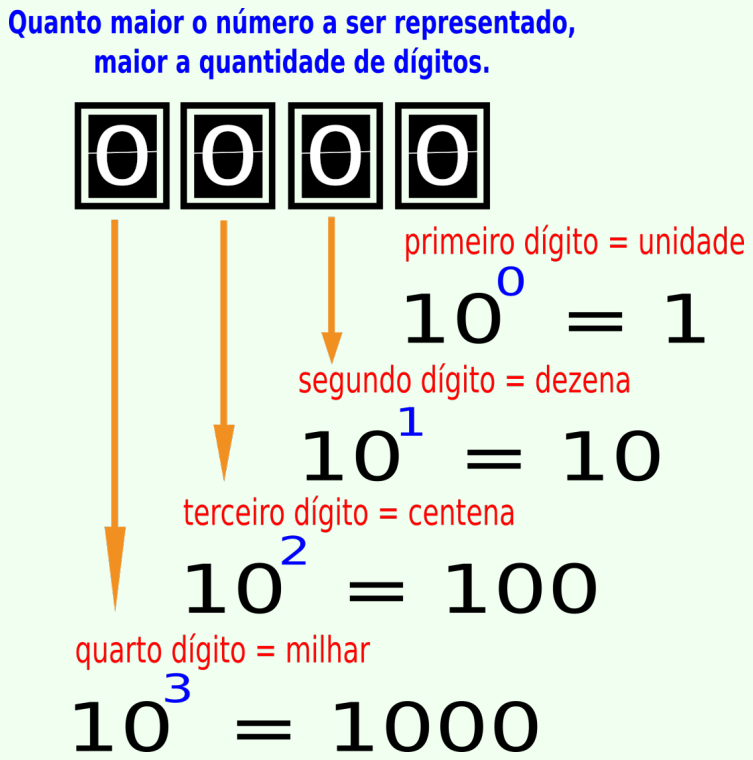
- Dígitos variam de 0 a 9
- Ao completar a contagem, aumenta-se um dígito à esquerda para representar a soma de potências de 10.



Sistemas de numeração

Sistema de base decimal

- Dígitos variam de 0 a 9
- Ao completar a contagem, aumenta-se um dígito à esquerda para representar a soma de potências de 10.



Teorema Fundamental da Numeração (TFN)

Exemplo: o número decimal 368

$$368_{10} = (368)_{10} = 3 \times 10^2 + 6 \times 10^1 + 8 \times 10^0$$

Atenção

Para sistemas de numeração de base ($>= 10$) é comum não usarmos parênteses e nem indicarmos a base pelo subíndice. Acima, poderíamos ter escrito apenas: 368 ao invés de 368_{10} ou $(368)_{10}$

OBSERVAÇÃO: Para o escopo dessa cadeira realizaremos somente operações básicas com números $\in \mathbb{N}$

TFN e o Sistema de numeração de base b

Genericamente

Dado um número natural $b > 1$ e o conjunto de símbolos $\{+, -, 0, 1, 2..., b-1\}$, a sequência de símbolos

$$(d_n, d_{n-1}...d_1, d_0, d_{-1}, d_{-2}...)_b$$

representa o número positivo

$$d_n \cdot b^n + d_{n-1} \cdot b^{n-1} + ... + d_0 \cdot b^0 + d_{-1} \cdot b^{-1} + d_{-2} \cdot b^{-2} + ...$$

Para representar números negativos usamos o símbolo $-$ a esquerda do numeral.

O sistema binário

Apenas 2 valores possíveis

- Dígitos variam de 0 a 1
- *Binary digit = "bit"*
- Ao completar a contagem, aumenta-se um dígito para representar a soma de potências de 2 (equivalentes ao número decimal).

Exemplo: o número binário 101 equivale ao número decimal 5

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

O sistema binário

Como a base binária tem menos algarismos disponíveis, necessitamos de mais dígitos para representar o número equivalente decimal

- Por exemplo: Quantos dígitos binários (bits) são necessários para representar o número decimal 5?
- Resposta: **3**
- Porque: $2^3 = 8$, sendo 3 o menor expoente capaz de representar este valor.

Veja a tabela:

Todos os valores binários possíveis com 3 dígitos

Decimal	0	1	2	3	4	5	6	7
Binário	000	001	010	011	100	101	110	111

Binary Digit

Atividades de fixação

- Bits
- Imagens
- Compressão
- Paridade

Outras bases notáveis

Base octal (8)

- Algarismos: 0 a 7
- Exemplo: $(163)_8$

Base Hexadecimal (16)

- Algarismos: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$
- Exemplo: $(4BF3)_{16}$

Confira esses valores!

Mudanças de base

Para decimal:

- utilize o TFN

De decimal para outra base:

- Preencher dígitos com as menores potências que compoem o número, ou
- Utilizar o resto da divisão sucessiva pela base.

De uma base qualquer para outra:

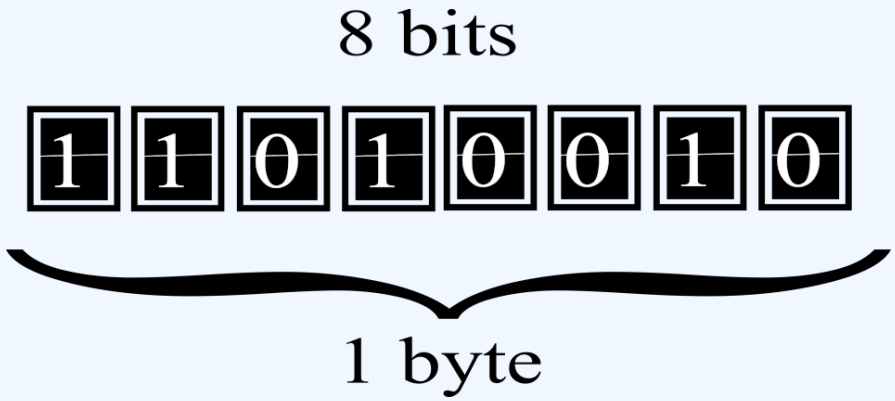
- Pode converter para decimal primeiro.
- ou utilize o código binário como intermediário.

Operações básicas

Algumas operações simples		
Decimal	Binário	Hexadecimal
5+6 = 11	101+110 = 1011	5+6 = B
14-3 = 11	1110-11 = 1011	E-3 = B

Medidas: bits, bytes...

ISO/IEC 2382-1:1993



- Bytes de oito bits também são chamados de octetos.
- Múltiplos de bits usam padrões prefixados, como quilobit (kb), megabit (Mb), gigabit (Gb), terabit (Tb) etc.
- A notação para bit utiliza um "b" minúsculo, ao contrário da notação para byte, que utiliza um "B" maiúsculo (kB, MB, GB, TB).

Padrões para representação

Caracteres

- ASCII - codificação
- UNICODE - mapeamento
- Pixels - imagens
- Videos? Áudios?

Exercícios

Questões para fixação

- A soma de dois números binários é 101000 e a diferença entre eles é igual a 1010. Quais são os números binários?
- Quantos bits são necessários para enumerar uma lista de 800 itens?
- Efetue as seguintes operações na base indicada para o resultado:

a) $7374_8 + 313202_4 = (\ )_{16}$
b) $532_6 + 101_8 = (\ )_{16}$
c) $1374_{10} + 6458_9 = (\ )_6$
d) $ACEF_{16} - 214_5 = (\ )_4$
e) $ECAE_{16} - 2D4F_{16} = (\ )_{16}$

Algoritmos

“Ciência da computação tem tanto a ver com o computador como a Astronomia com o telescópio, a Biologia com o microscópio, ou a Química com os tubos de ensaio. A Ciência não estuda ferramentas, mas o que fazemos e o que descobrimos com elas." ”

Edsger Dijkstra.

Dijkstra foi o principal contribuidor no final dos anos 1950 para o desenvolvimento do ALGOL, uma linguagem de programação de alto nível, modelo de clareza e rigor matemático. Ele é um dos maiores expoentes na arte e ciência das linguagens de programação, e contribuiu enormemente para a compreensão de sua estrutura, representação, e implementação. Seus quinze anos de publicações vão de artigos teóricos na teoria dos grafos até manuais, textos expositores, e filosofia contemplativa no campo das linguagens de programação.

Algoritmo

- O termo algoritmo tem origem na Matemática. A palavra "algoritmo" vem da transcrição europeia do nome do astrônomo e matemático do século IX *Al Khwarizmi*, de Bagdá, que catalogou e aperfeiçoou esses métodos, sendo o inventor de muitos deles.
- Sua compilação de algoritmos, *Hisab al-jabr w'al-muqabala*, é considerada o primeiro tratado de Álgebra da história. O termo "álgebra", inclusive, vem da expressão *al-jabr* contida em seu título.

Algoritmo

Algoritmo

Algoritmo

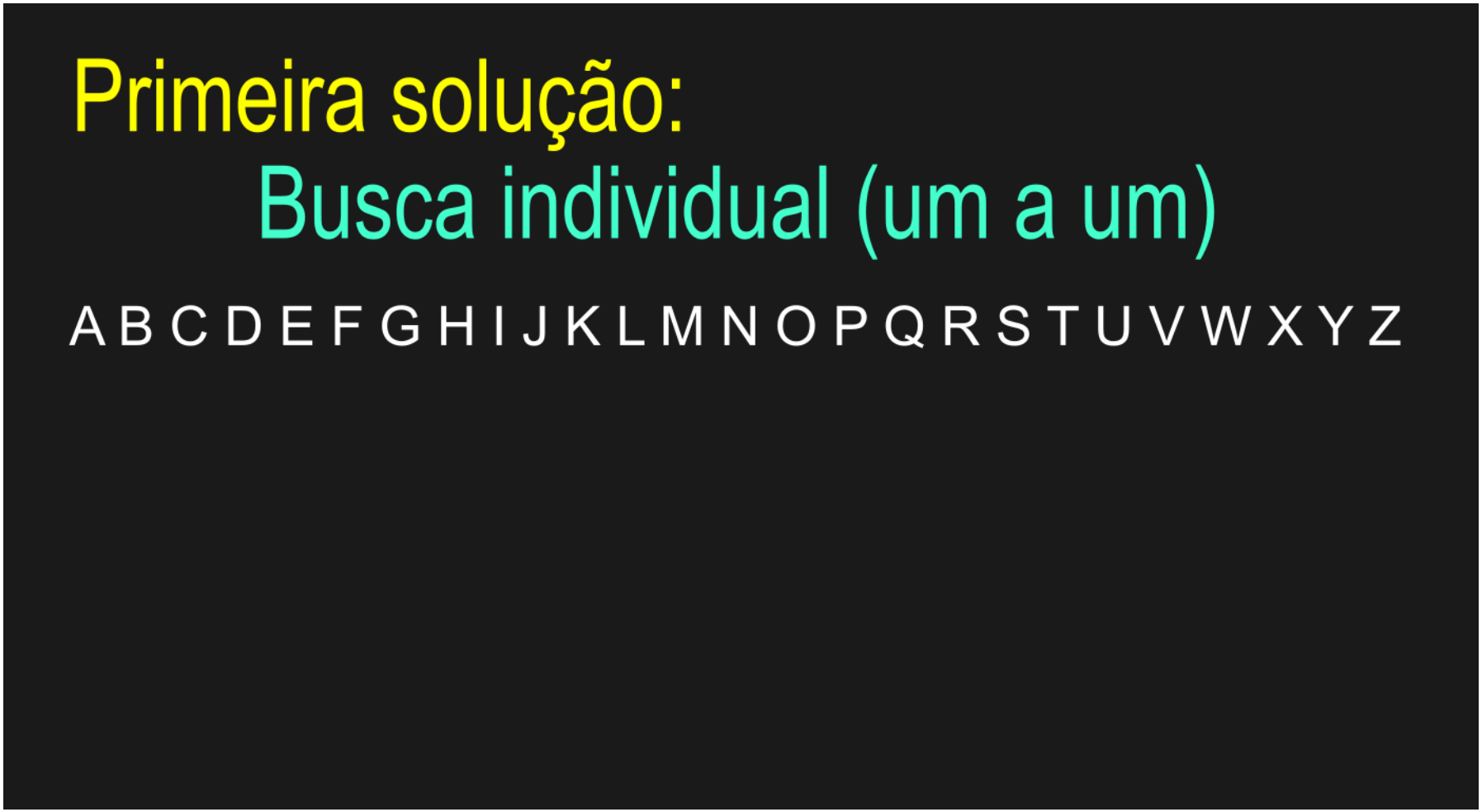
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Procurar determinado
nome
em uma lista telefônica
que está ordenada alfabeticamente

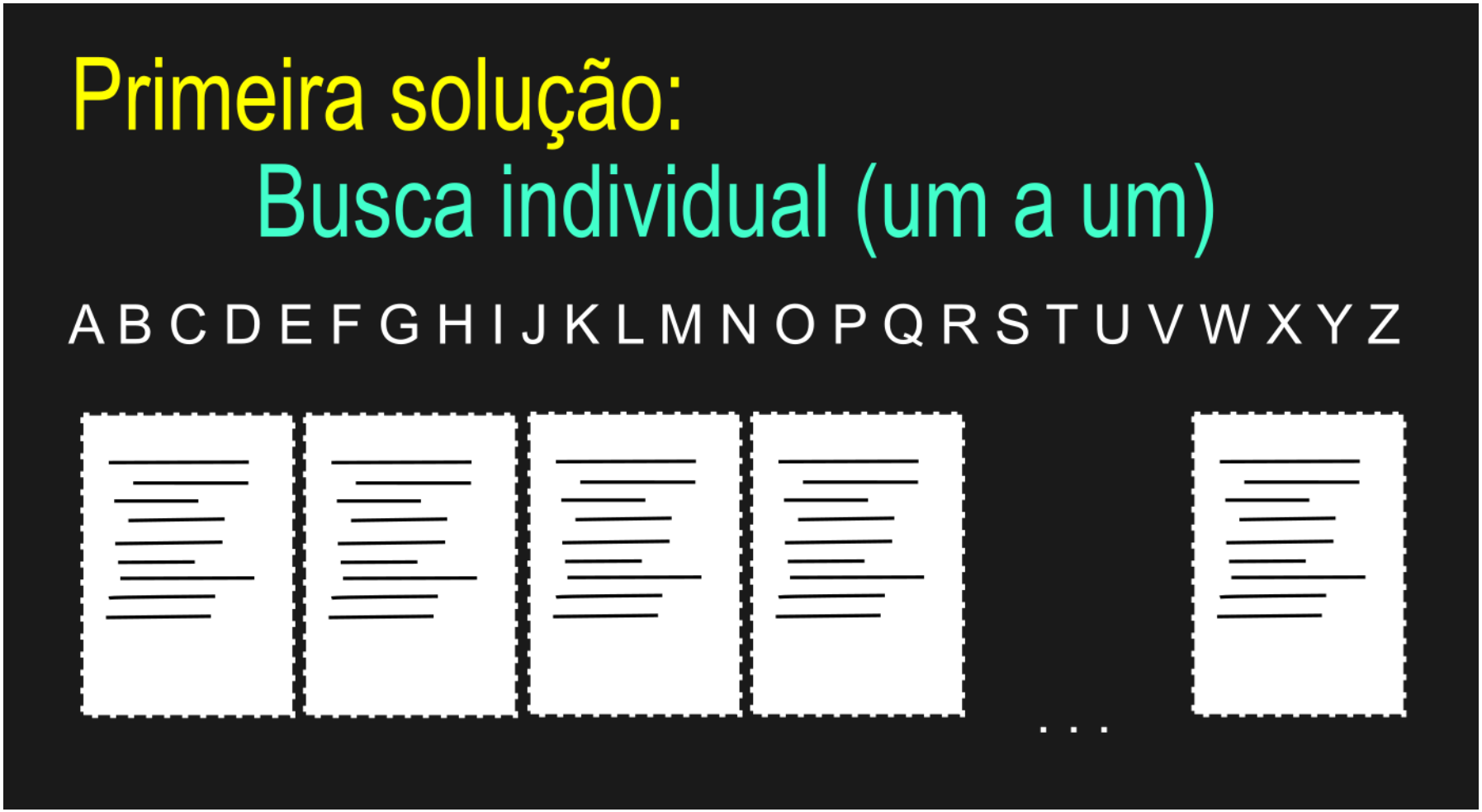
Algoritmo

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Hipoteticamente a lista tem
2.000 páginas

Algoritmo


Primeira solução:
Busca individual (um a um)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

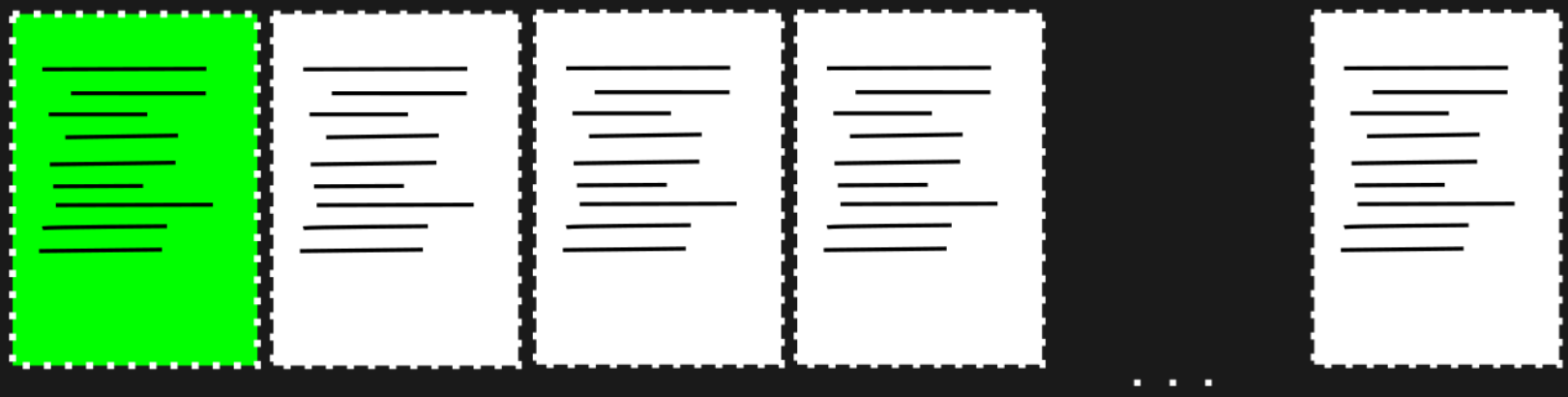
Algoritmo


Primeira solução:
Busca individual (um a um)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
. . .

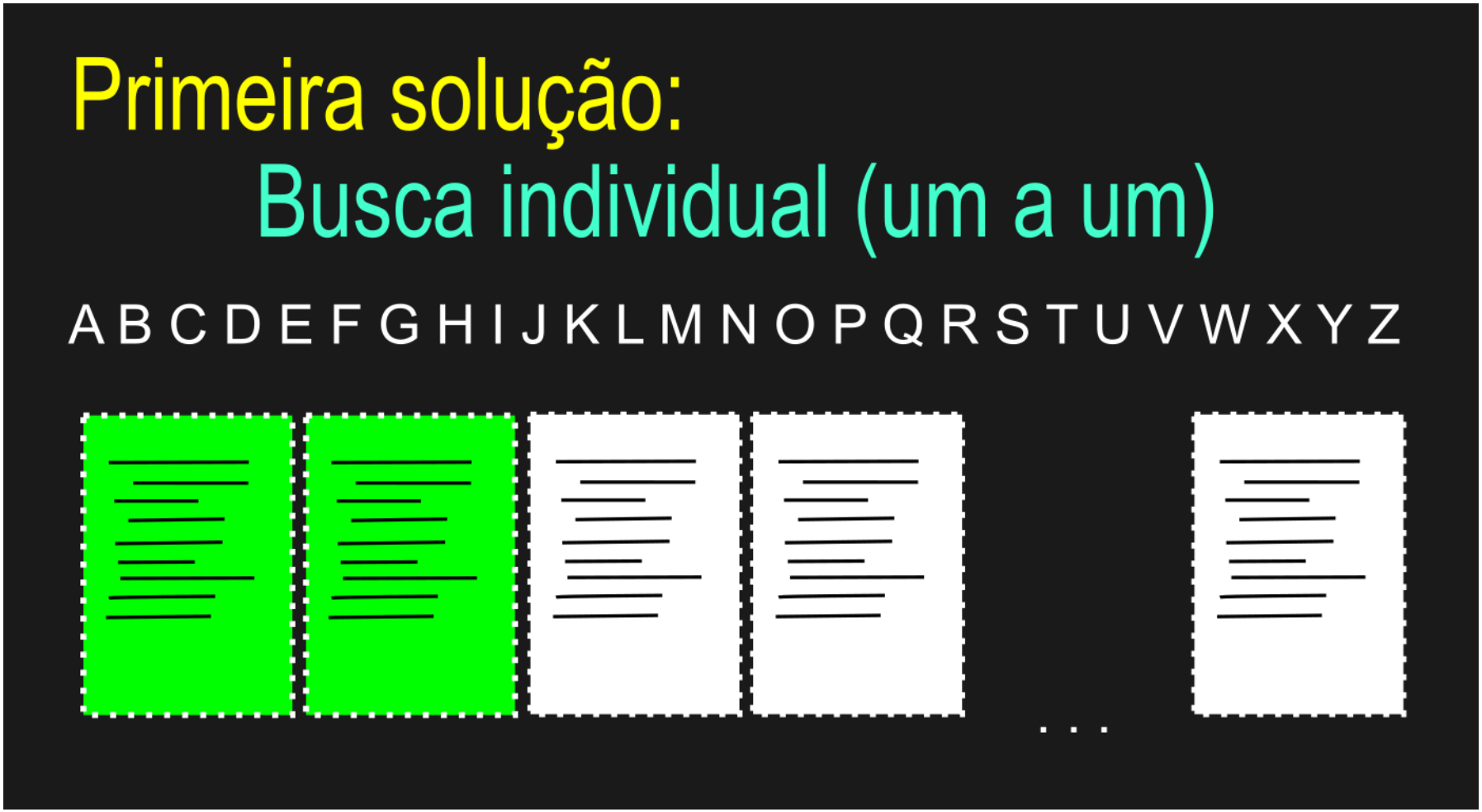
Algoritmo

Primeira solução:
Busca individual (um a um)

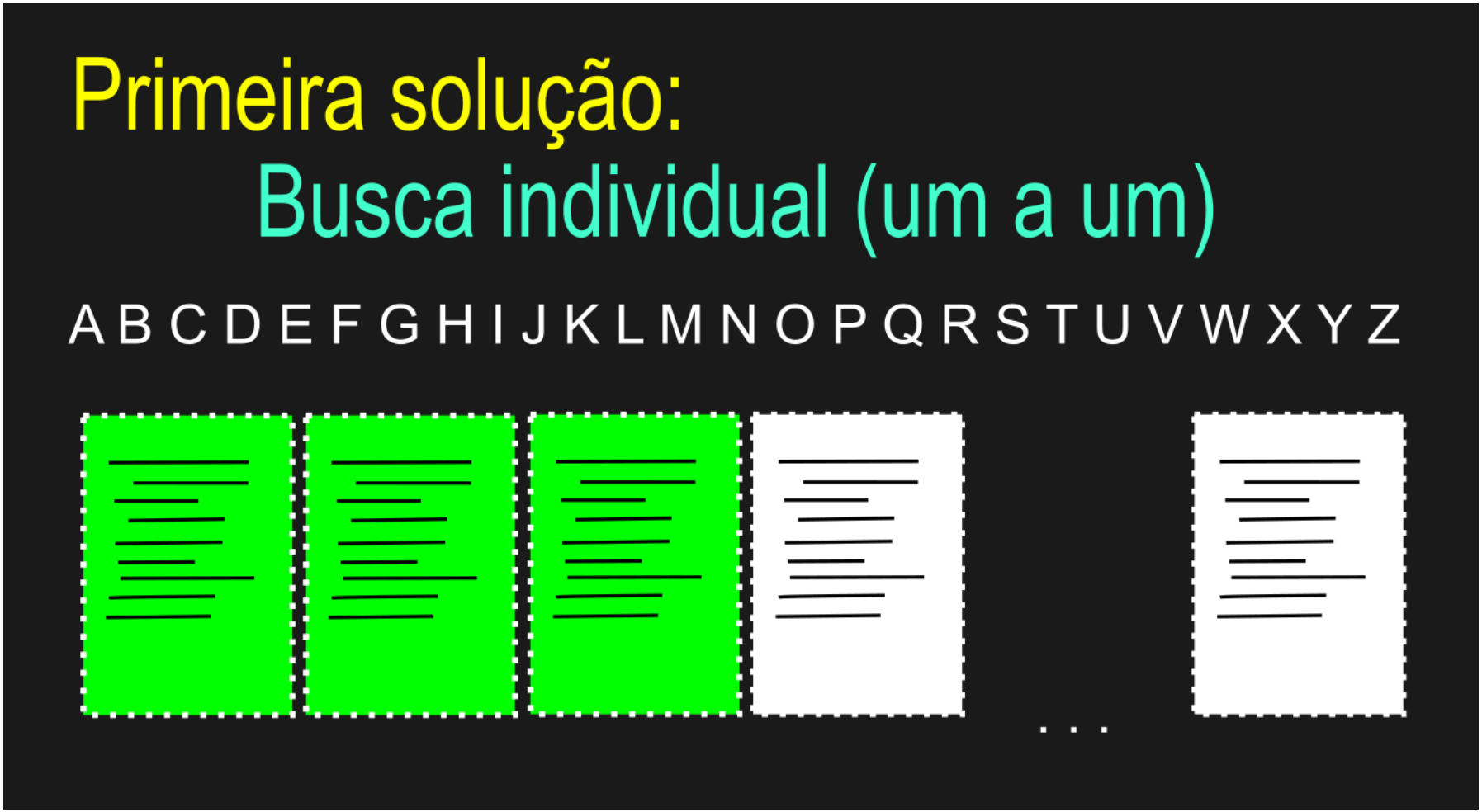
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



Algoritmo


Primeira solução:
Busca individual (um a um)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
. . .

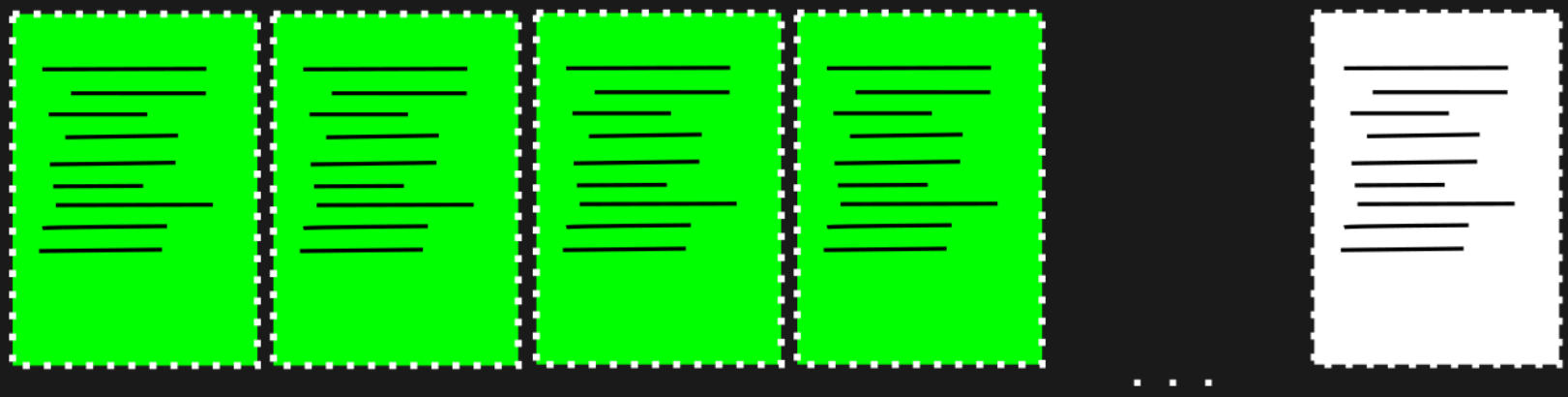
Algoritmo


Primeira solução:
Busca individual (um a um)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
. . .

Algoritmo

Primeira solução:
Busca individual (um a um)

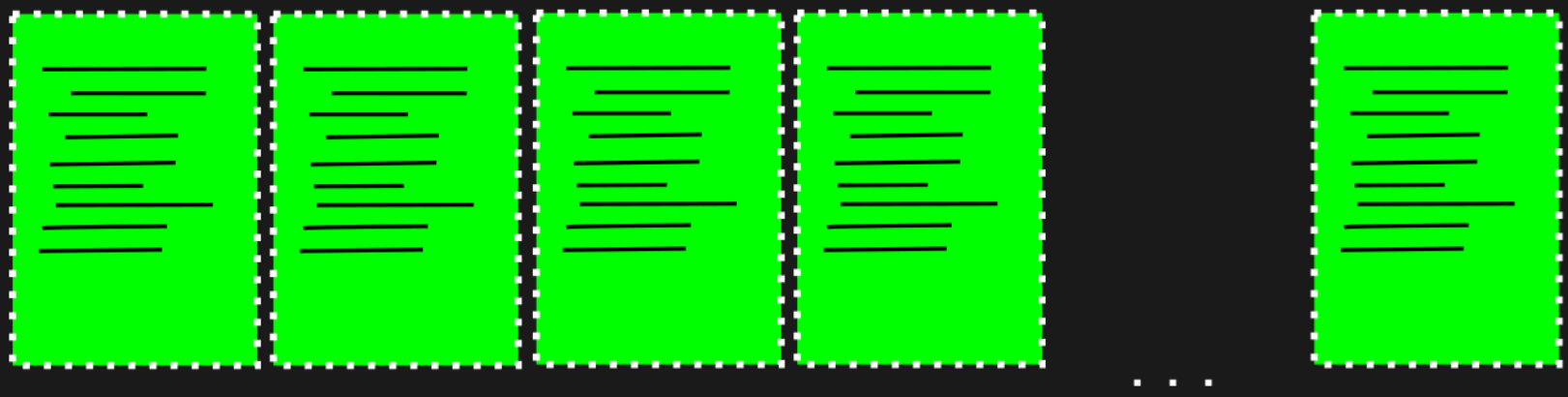
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



Algoritmo

Primeira solução:
Busca individual (um a um)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

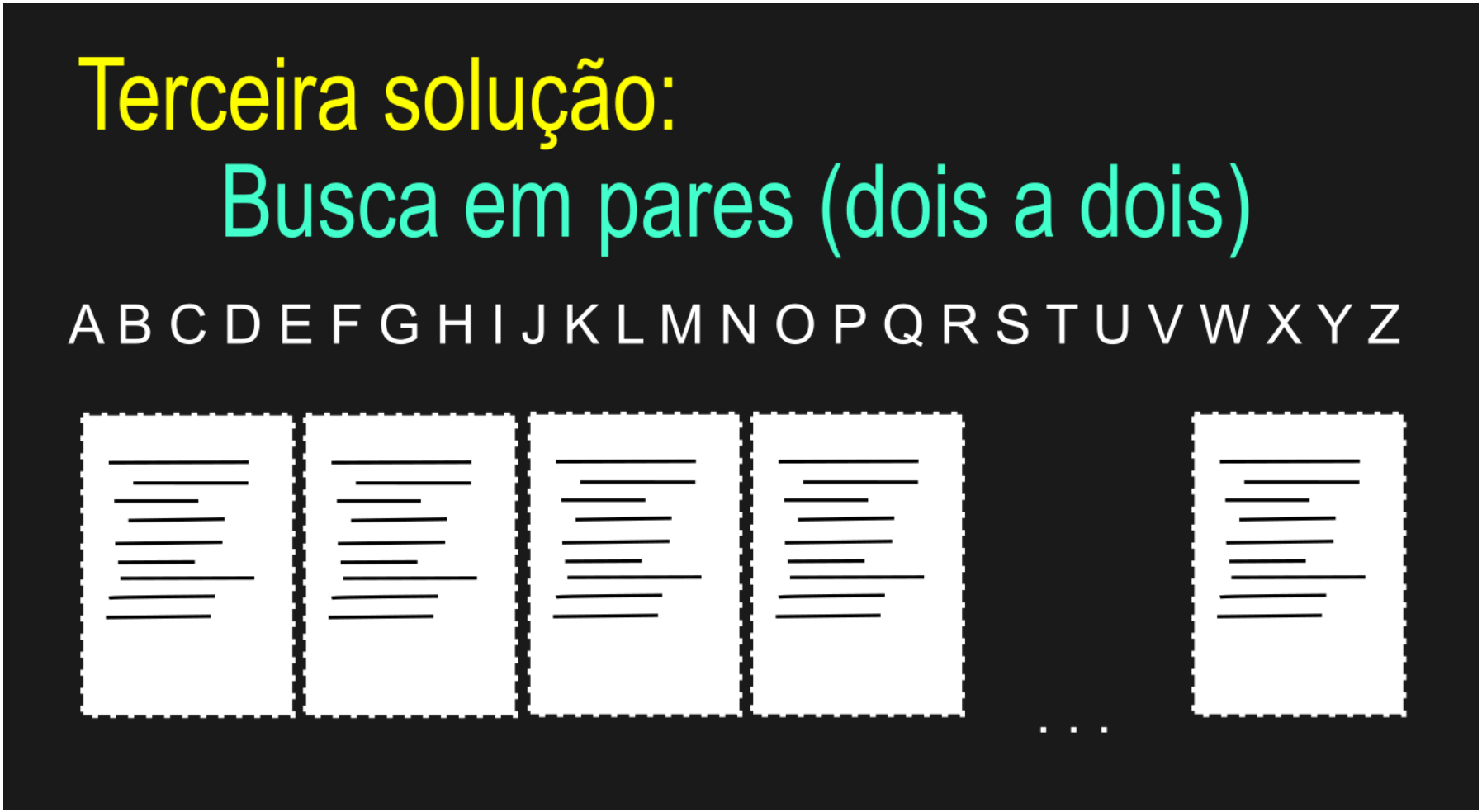


Algoritmo

Terceira solução:
Busca em pares (dois a dois)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

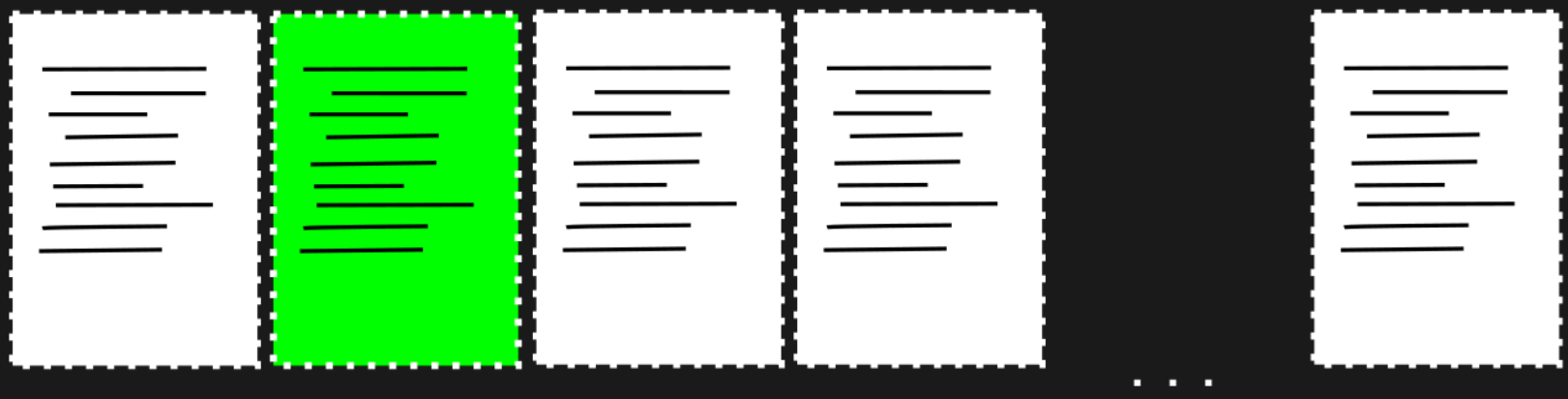
Algoritmo


Terceira solução:
Busca em pares (dois a dois)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
. . .

Algoritmo

Terceira solução:
Busca em pares (dois a dois)

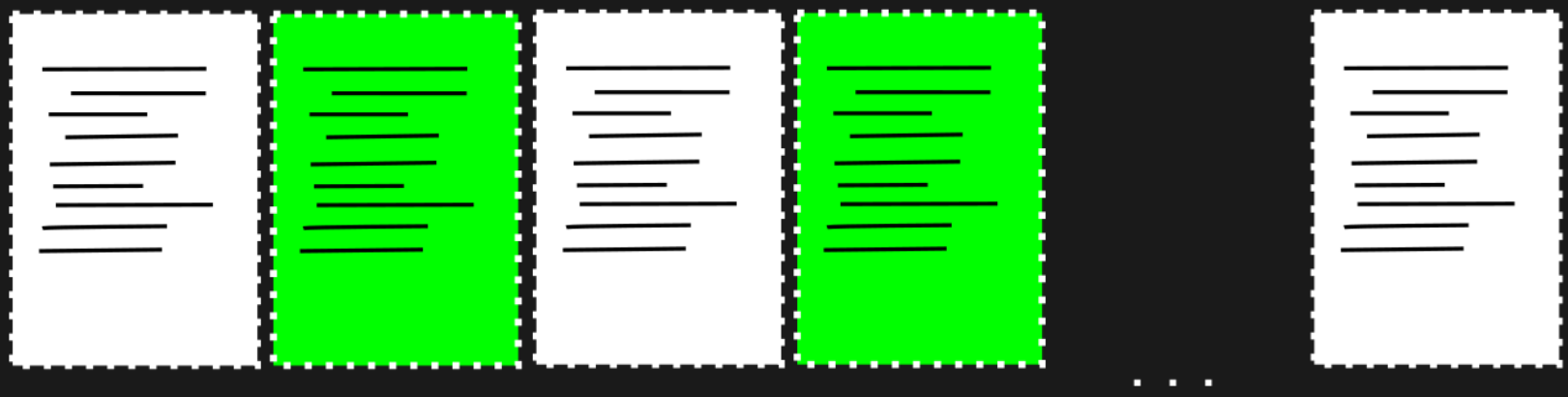
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



Algoritmo

Terceira solução:
Busca em pares (dois a dois)

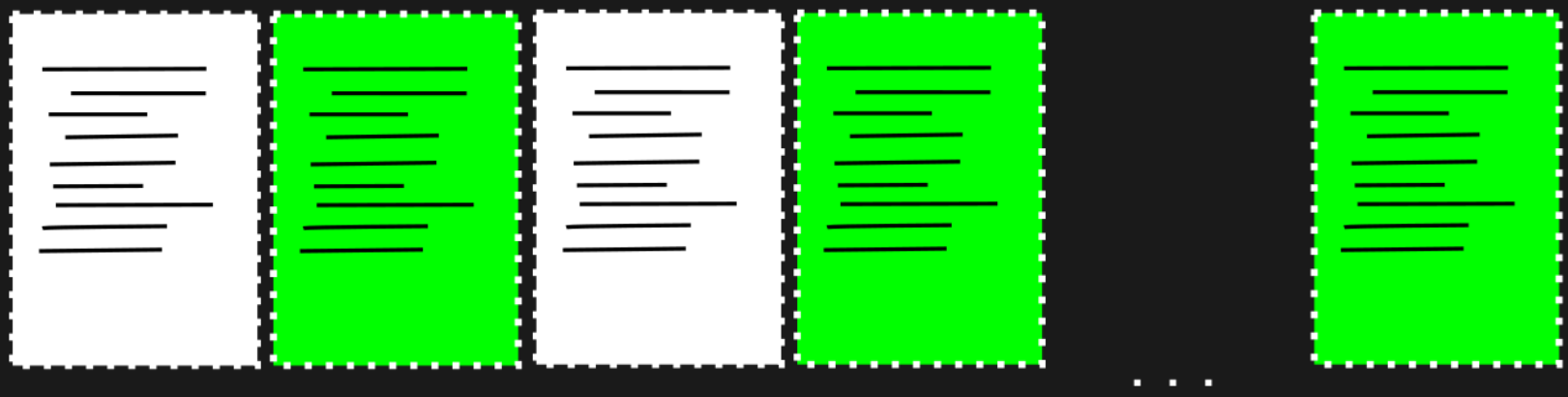
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



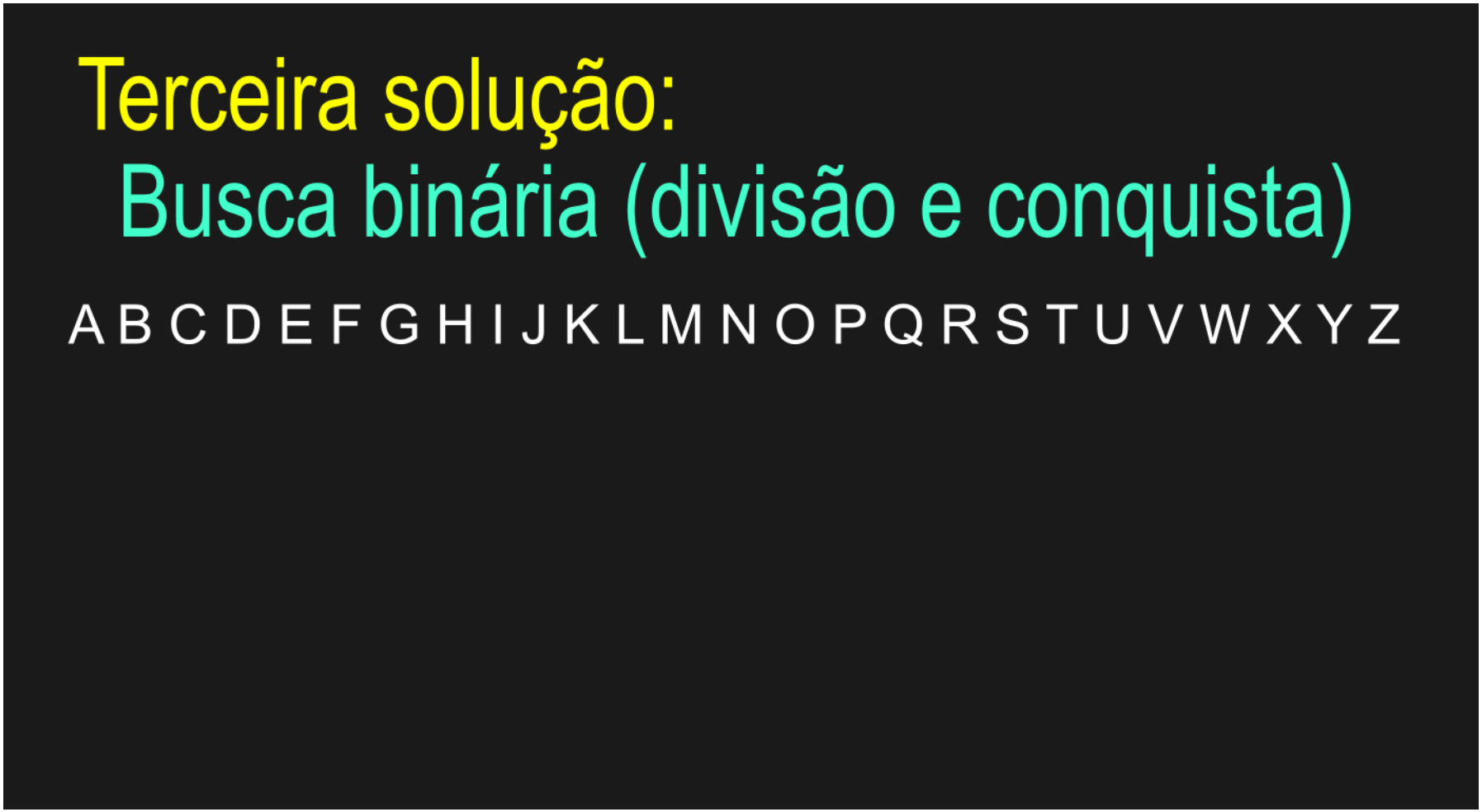
Algoritmo

Terceira solução:
Busca em pares (dois a dois)

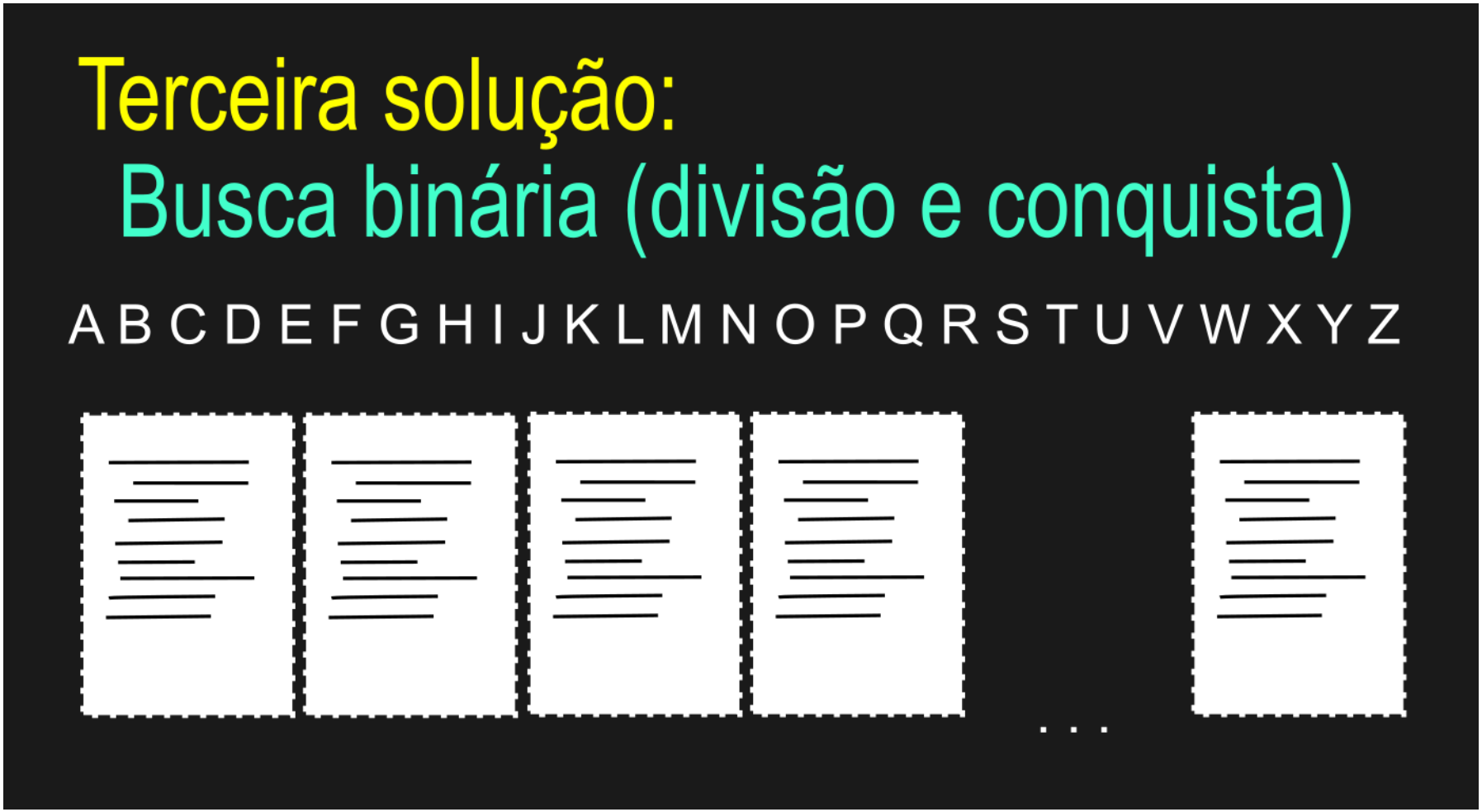
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



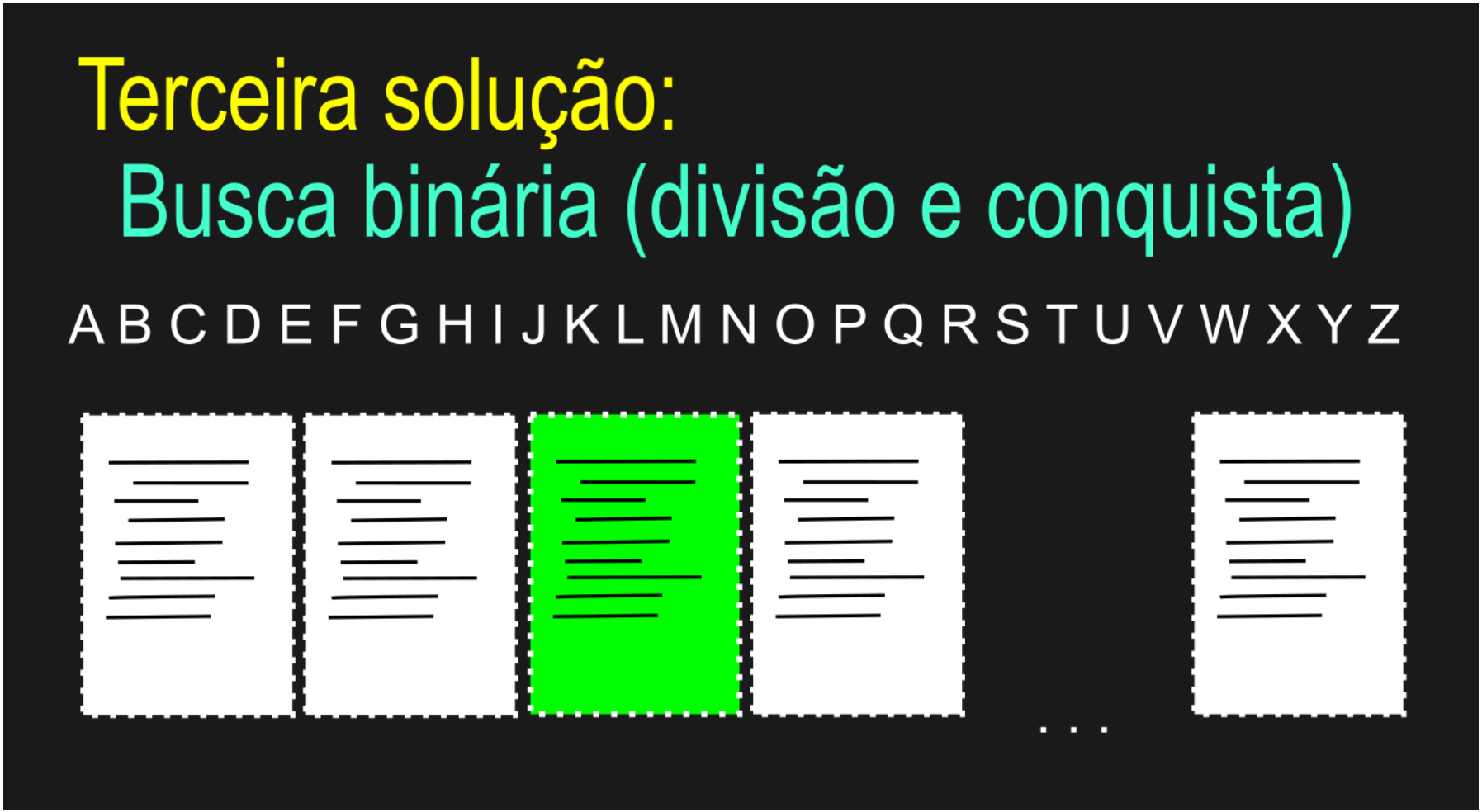
Algoritmo


Terceira solução:
Busca binária (divisão e conquista)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Algoritmo


Terceira solução:
Busca binária (divisão e conquista)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
. . .

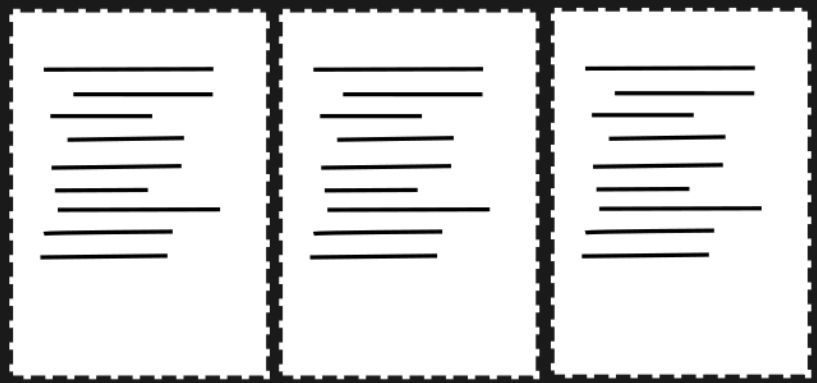
Algoritmo


Terceira solução:
Busca binária (divisão e conquista)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
. . .

Algoritmo

Terceira solução:
Busca binária (divisão e conquista)

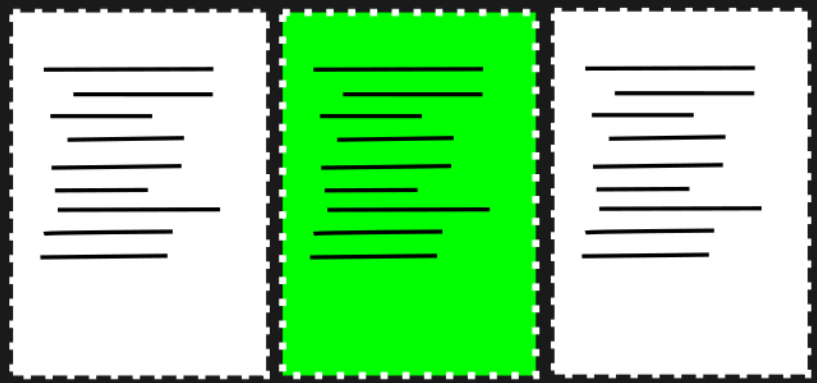
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



Algoritmo

Terceira solução:
Busca binária (divisão e conquista)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



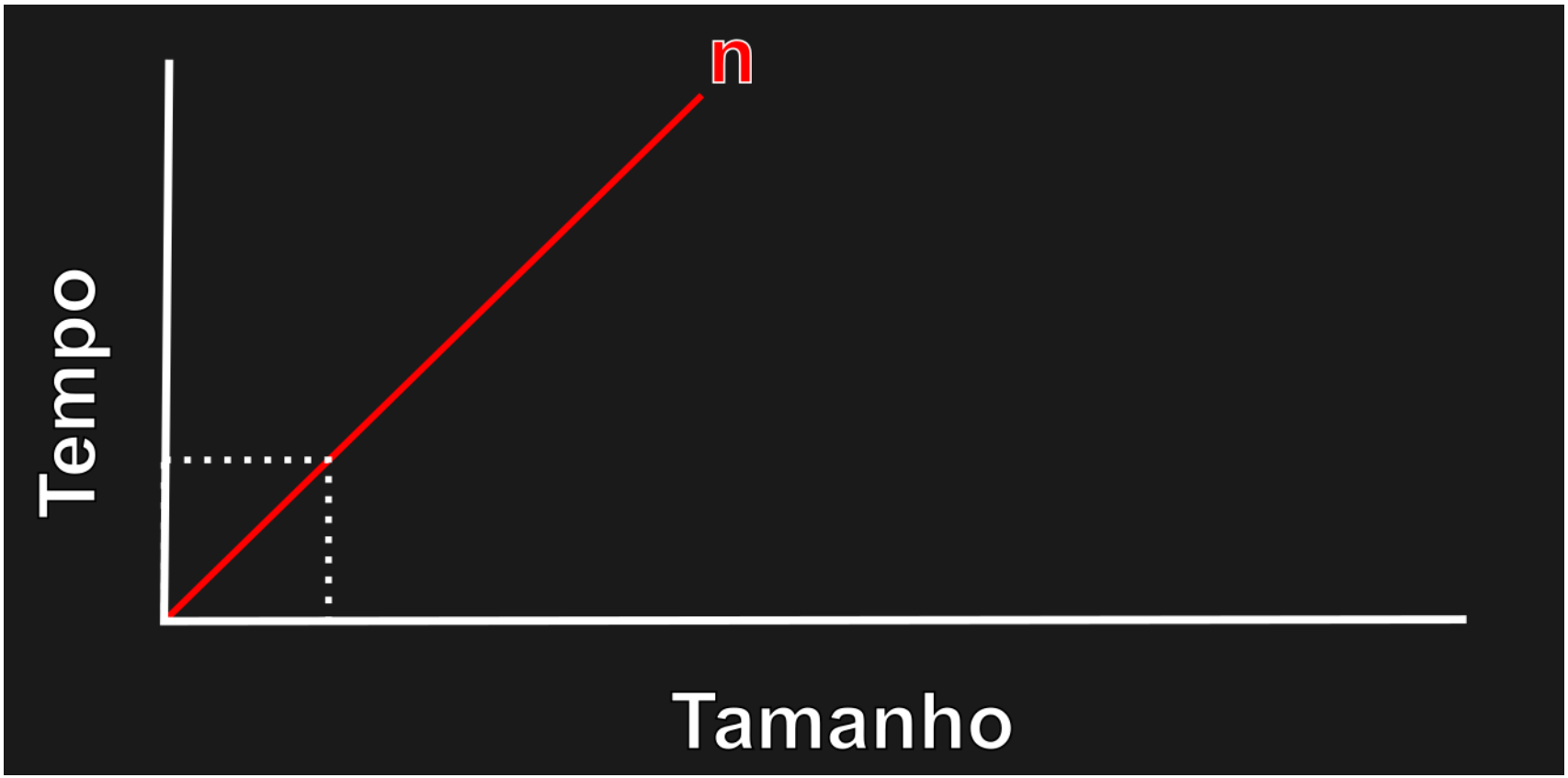
Algoritmo

Terceira solução:
Busca binária (divisão e conquista)

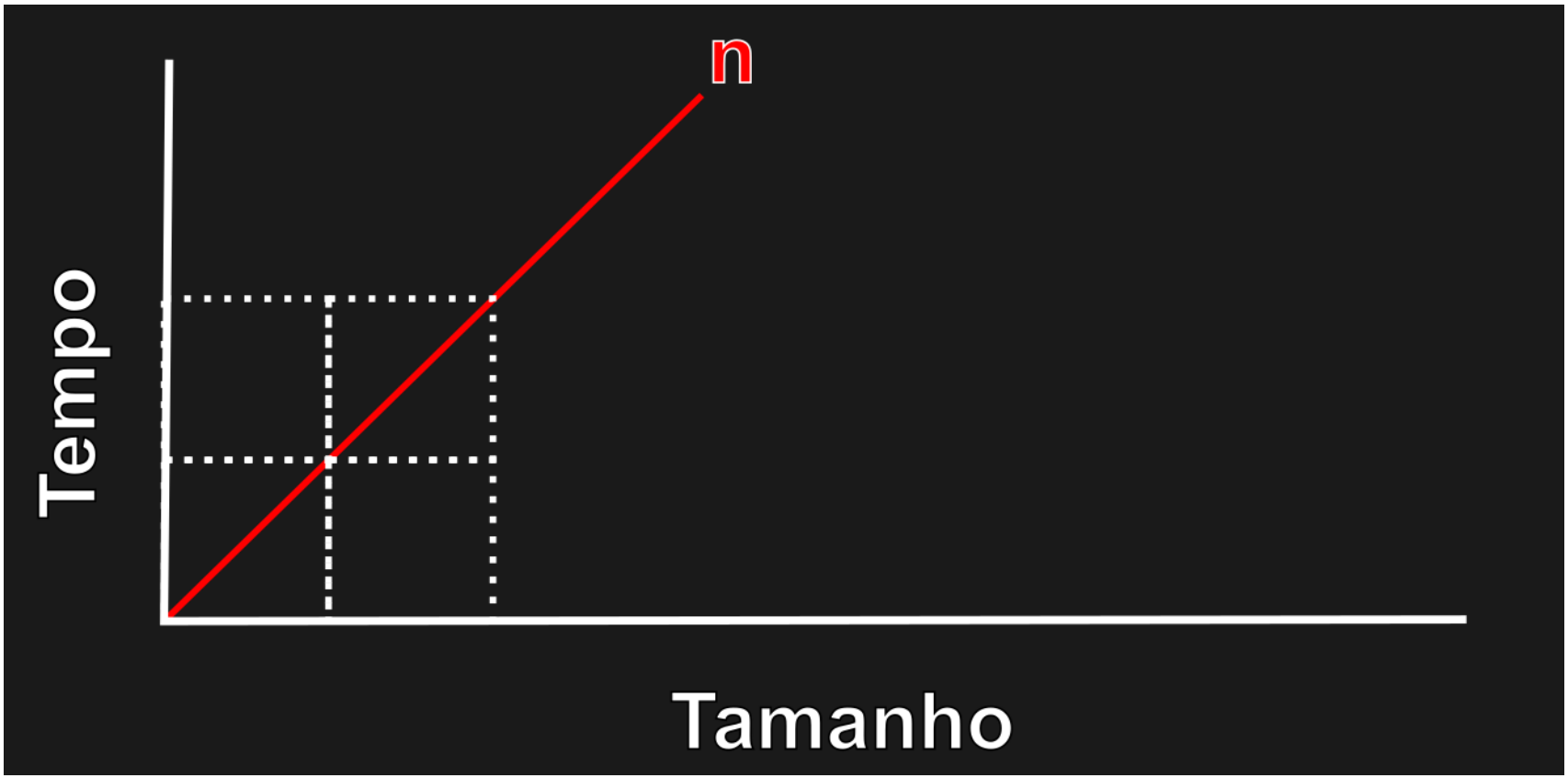
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Algoritmo

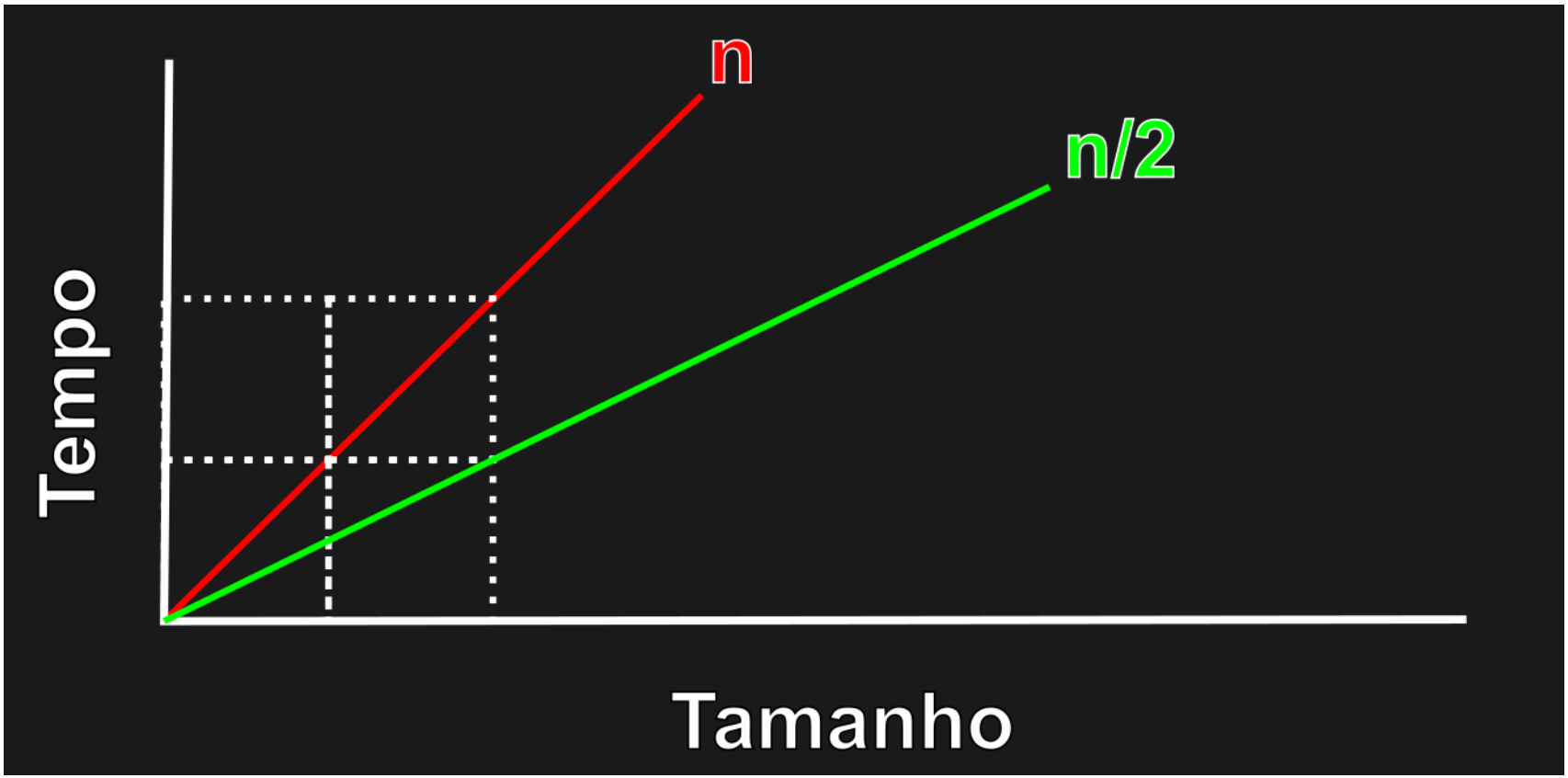
Algoritmo



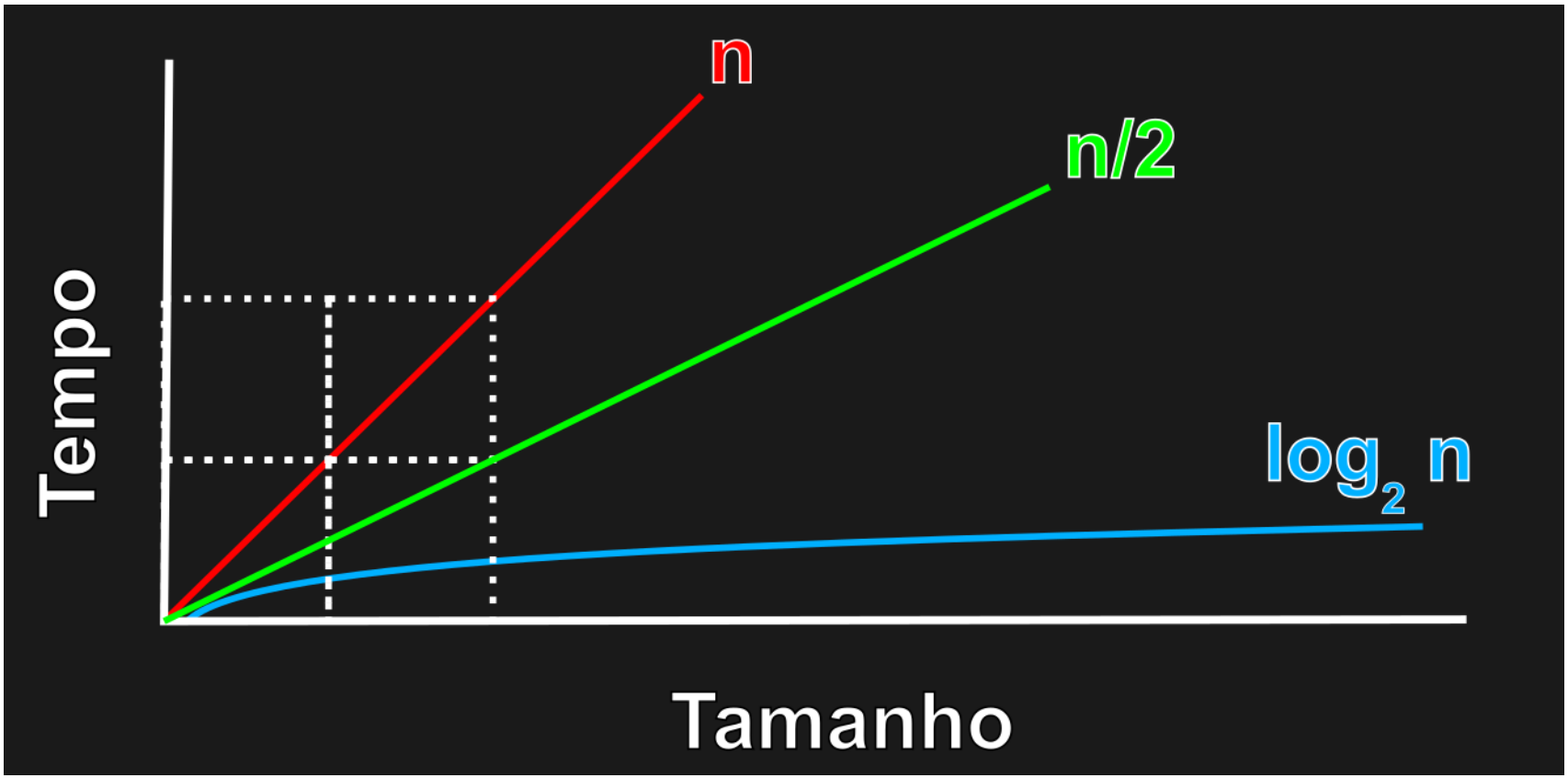
Algoritmo



Algoritmo



Algoritmo



Algoritmo

Conjunto de instruções que devem ser feitas para realizar uma tarefa.

Características:

- Finitude;
- Passos não podem ser ambíguos;
- Entradas (informações que são fornecidas antes do algoritmo iniciar);
- Saídas (resultado final do algoritmo);
- Efetividade: todas as operações devem ser suficientemente básicas de modo que possam ser executadas com precisão em um tempo finito.

Algoritmo

- Os algoritmos costumam aparecer **codificados** em notações complexas chamadas linguagens de programação, e são os elementos constitutivos dos programas.
- A especificação de passos do algoritmo na linguagem natural é chamada de **pseudo-código**.

Pseudo-código

Procurar nome na lista telefônica

1. Pegar a lista telefônica

Pseudo-código

Procurar nome na lista telefônica

1. Pegar a lista telefônica
2. Abrir no meio da lista

Pseudo-código

Procurar nome na lista telefônica

1. Pegar a lista telefônica
2. Abrir no meio da lista
3. Olhar para a página

Pseudo-código

Procurar nome na lista telefônica

1. Pegar a lista telefônica
2. Abrir no meio da lista
3. Olhar para a página
4. Se o nome está na página
   - Encontrou!

Pseudo-código

Procurar nome na lista telefônica

1. Pegar a lista telefônica
2. Abrir no meio da lista
3. Olhar para a página
4. Se o nome está na página
   - Encontrou!
5. senão se o nome está antes
   - abra no meio da metade anterior
   - Vá para linha 3

Pseudo-código

Procurar nome na lista telefônica

1. Pegar a lista telefônica
2. Abrir no meio da lista
3. Olhar para a página
4. Se o nome está na página
   - Encontrou!
5. senão se o nome está antes
   - abra no meio da metade anterior
   - Vá para linha 3
6. senão se o nome está depois
   - abra no meio da metade posterior
   - Vá para linha 3

Pseudo-código

Procurar nome na lista telefônica

1. Pegar a lista telefônica
2. Abrir no meio da lista
3. Olhar para a página
4. Se o nome está na página
   - Encontrou!
5. senão se o nome está antes
   - abra no meio da metade anterior
   - Vá para linha 3
6. senão se o nome está depois
   - abra no meio da metade posterior
   - Vá para linha 3
7. senão
   - termina a busca (não encontrou!)

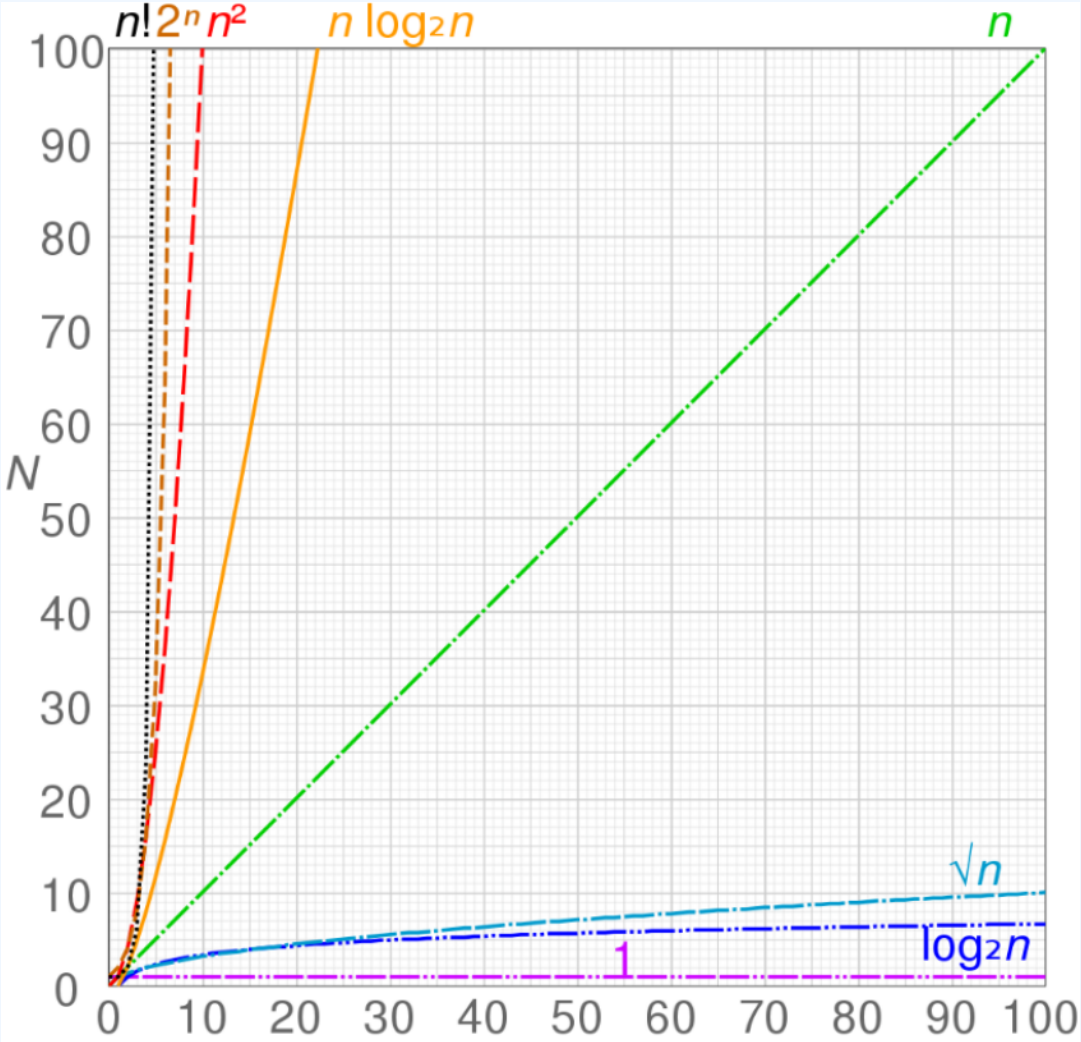
Pseudo-código

Estruturas importantes

- Ações (funções)
- Desvios condicionais
- Expressão booleana (lógica)
- Repetições

Exemplos de algoritmos

Notação big-O



Exercício

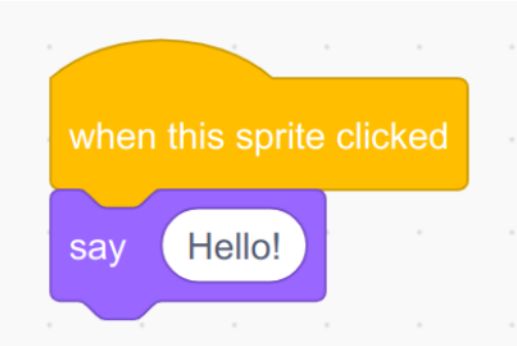
Pseudo-código simplificado

- Considere um estoque de laranjas guardadas em frigorífico já lavadas
- Considere que espremer exatamente 5 laranjas é suficiente para um copo de 200ml de suco;
- Elabore um pseudo-código para a produção de 3 litros de suco de laranja.
- Identifique a complexidade de tempo deste algoritmo

O programa

- Programar é basicamente construir algoritmos de forma sistemática.
- O aspecto fundamental nos programas é que eles são formulações concretas de algoritmos abstratos. Estas formulações são baseadas em representações e estruturas específicas de dados.
- Tais estruturas são representações das informações do problema a ser resolvido.
- Formular algoritmo e definir estruturas de dados apropriadas são tarefas intimamente ligadas

Ilustração - Scratch (MIT)



- Vamos aprender alguns conceitos aqui através de exemplos de aplicações no *Scratch*