



# Estrutura de Dados I

## Busca binária e interpolada

Bruno Prado

Departamento de Computação / UFS

# Introdução

- ▶ Problema de busca
  - ▶ Consiste em aplicar de um algoritmo de busca para encontrar um determinado item que pode estar armazenado na estrutura de dados

91	3	43	...	23	54	32
0	1	2		$n-3$	$n-2$	$n-1$

Dados não ordenados  $\rightarrow O(n)$

# Introdução

- ▶ Informações armazenadas com ordenação
  - ▶ Ata de presença
  - ▶ Palavras do dicionário
  - ▶ Catálogo telefônico
  - ▶ ...

# Introdução

- ▶ Informações armazenadas com ordenação
  - ▶ Ata de presença
  - ▶ Palavras do dicionário
  - ▶ Catálogo telefônico
  - ▶ ...

Como seria buscar estas informações  
sem ordenação?

# Introdução

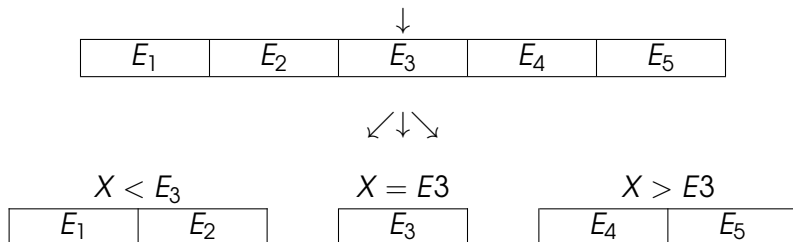
- ▶ Custo da ordenação x Eficiência de busca
  - ▶ A complexidade de algoritmos eficientes de ordenação pode variar de  $O(n)$  até  $O(n \log n)$ , dependendo das características dos dados
  - ▶ O princípio de operação de uma busca mais eficiente é aproveitar a ordenação dos elementos para reduzir o número de comparações

# Introdução

- ▶ Custo da ordenação x Eficiência de busca
  - ▶ A complexidade de algoritmos eficientes de ordenação pode variar de  $O(n)$  até  $O(n \log n)$ , dependendo das características dos dados
  - ▶ O princípio de operação de uma busca mais eficiente é aproveitar a ordenação dos elementos para reduzir o número de comparações
- ▶ Técnicas de busca em vetores ordenados
  - ▶ Binária
  - ▶ Interpolada

# Busca Binária

- ▶ Como funciona a busca binária?
  - ▶ Cada passo o vetor é dividido em duas partes
  - ▶ É verificado qual parte contém o elemento  $X$



# Busca Binária

- ▶ Como funciona a busca binária?
  - ▶ Caso base 1: o elemento  $X$  está no meio do vetor
  - ▶ O índice do elemento é retornado

$$X = E_3$$

$E_3$
-------



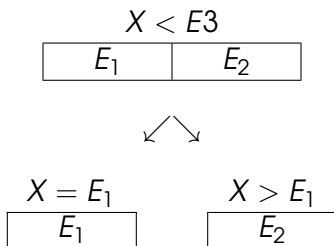
# Busca Binária

- ▶ Como funciona a busca binária?
  - ▶ Caso base 2: o elemento  $X$  não está no vetor
  - ▶ É retornado um índice negativo

$X < E_1$	$X > E_5$
$E_1$	$E_5$

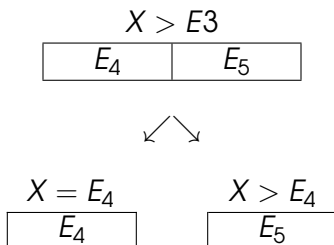
# Busca Binária

- ▶ Como funciona a busca binária?
  - ▶ Caso recursivo 1: o elemento  $X$  pode estar armazenado na metade inferior
  - ▶ O vetor é dividido novamente em duas partes para realização de uma nova busca considerando somente metade dos dados



# Busca Binária

- ▶ Como funciona a busca binária?
  - ▶ Caso recursivo 2: o elemento  $X$  pode estar armazenado na metade superior
  - ▶ O vetor é dividido novamente em duas partes para realização de uma nova busca considerando somente metade dos dados



# Busca Binária

- ▶ Busca em vetores com ordenação
  - ▶ Parâmetro de busca: 13
  - ▶ É feito o cálculo do índice do elemento pivô que divide o vetor em duas partes e a comparação do valor deste elemento com o valor procurado

2	3	5	8	13	21	34
0	1	2	3	4	5	6

↑

$$i = \frac{0 + 6}{2} = 3$$

$$V[i] \leftrightarrow 13$$

# Busca Binária

- ▶ Busca em vetores com ordenação
  - ▶ Parâmetro de busca: 13
  - ▶ É verificado que o elemento procurado pode estar armazenado na metade superior do vetor, uma vez que o elemento central é menor do que o pivô

2	3	5	8	13	21	34
0	1	2	3	4	5	6
			↑			

$$i = \frac{0 + 6}{2} = 3$$

$$V[i] > 13$$

# Busca Binária

- ▶ Busca em vetores com ordenação
  - ▶ Parâmetro de busca: 13
  - ▶ É feito o cálculo do índice do elemento pivô que divide o vetor em duas partes e a comparação do valor deste elemento com o valor procurado

2	3	5	8	13	21	34
0	1	2	3	4	5	6

↑

$$i = \frac{4 + 6}{2} = 5$$

$$V[i] \leftrightarrow 13$$

# Busca Binária

- ▶ Busca em vetores com ordenação
  - ▶ Parâmetro de busca: 13
  - ▶ É verificado que o elemento procurado pode estar armazenado na metade inferior do vetor, uma vez que o elemento central é menor do que o pivô

2	3	5	8	13	21	34
0	1	2	3	4	5	6
					↑	

$$i = \frac{4 + 6}{2} = 5$$

$$V[i] > 13$$

# Busca Binária

- ▶ Busca em vetores com ordenação
  - ▶ Parâmetro de busca: 13
  - ▶ É feito o cálculo do índice do elemento pivô que divide o vetor em duas partes e a comparação do valor deste elemento com o valor procurado

2	3	5	8	13	21	34
0	1	2	3	4	5	6

↑

$$i = \frac{4 + 4}{2} = 4$$

$$V[i] \leftrightarrow 13$$



# Busca Binária

- ▶ Busca em vetores com ordenação
  - ▶ Parâmetro de busca: 13
  - ▶ O elemento procurado é encontrado na posição 4 e o valor de seu índice  $i$  é retornado

2	3	5	8	13	21	34
0	1	2	3	4	5	6

↑

$$i = \frac{4 + 4}{2} = 4$$

$$V[i] = 13$$

# Busca Binária

## ► Implementação recursiva em C

```
// Busca binária recursiva
int busca_bin_r(int* vetor, int inf, int sup, int valor) {
    int i = (sup + inf) / 2;
    // Casos bases
    if(sup < inf)
        return -1;
    else if(vetor[i] == valor)
        return i;
    // Casos recursivos
    else if(vetor[i] < valor)
        return busca_bin_r(vetor, i + 1, sup, valor);
    else
        return busca_bin_r(vetor, inf, i - 1, valor);
}
```

# Busca Binária

## ► Implementação iterativa em C

```
// Busca binária iterativa
int busca_bin_i(int* vetor, int tamanho, int valor) {
    int inf = 0, sup = tamanho - 1;
    int i = (sup + inf) / 2;
    // Busca iterativa
    while(sup >= inf && vetor(i) != valor) {
        if(vetor(i) > valor)
            sup = i - 1;
        else
            inf = i + 1;
        i = (sup + inf) / 2;
    }
    return (vetor(i) == valor) ? (i) : (-1);
}
```

# Busca Binária

- ▶ Análise de complexidade
  - ▶ Princípio de funcionamento

$$\sum_{i=1}^k \frac{n}{2^i} = \frac{n}{2^1} + \frac{n}{2^2} + \dots + \frac{n}{2^k}$$

$$\frac{n}{2^k} = 1 \rightarrow 2^k = n \rightarrow k = \log_2 n$$

# Busca Binária

- ▶ Análise de complexidade
  - ▶ Princípio de funcionamento

$$\sum_{i=1}^k \frac{n}{2^i} = \frac{n}{2^1} + \frac{n}{2^2} + \dots + \frac{n}{2^k}$$

$$\frac{n}{2^k} = 1 \rightarrow 2^k = n \rightarrow k = \log_2 n$$

- ▶ Espaço
  - ▶ Iterativo  $O(1)$
  - ▶ Recursivo  $O(\log_2 n)$
- ▶ Tempo  $O(\log_2 n)$

# Busca Interpolada

- ▶ O que é uma busca interpolada?
  - ▶ É um algoritmo de busca que calcula a posição do elemento procurado baseando-se em informações sobre o conjunto de dados
  - ▶ Esta técnica é geralmente utilizada por pessoas na busca por informações com ordenação, como em agendas, dicionários ou listas telefônicas

# Busca Interpolada

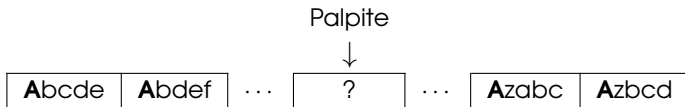
- ▶ Como buscar de forma interpolada?
  - ▶ É preciso que a estrutura de dados esteja ordenada e que a distribuição dos elementos seja conhecida

1	9	17	25	33	41	49	57
0	1	2	3	4	5	6	7

$$\begin{aligned}busca\_int(x) &= \left\lfloor 0 + \left( \frac{7-0}{57-1} \right) \times (x-1) \right\rfloor \\ &= \left\lfloor \frac{7x}{56} \right\rfloor \\ &= \left\lfloor \frac{x}{8} \right\rfloor\end{aligned}$$

# Busca Interpolada

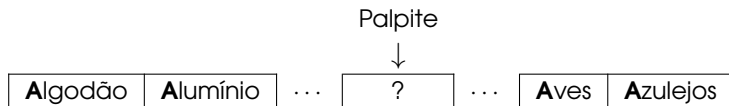
- ▶ Como buscar de forma interpolada?
  - ▶ Lista telefônica: automóveis
  - ▶ A distribuição dos nomes é ilustrada na borda das páginas e sua espessura indica quantos contatos estão armazenados





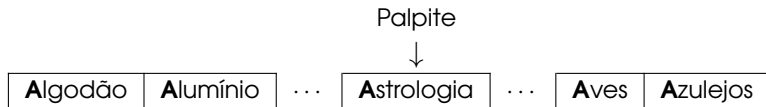
# Busca Interpolada

- ▶ Como buscar de forma interpolada?
  - ▶ Lista telefônica: automóveis
  - ▶ Baseando-se na ordenação alfabética e na quantidade de nomes, é feito o cálculo probabilístico do índice para o termo procurado



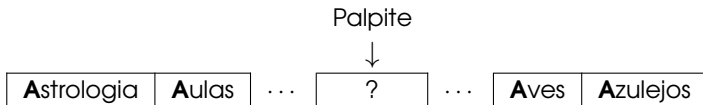
# Busca Interpolada

- ▶ Como buscar de forma interpolada?
  - ▶ Lista telefônica: automóveis
  - ▶ O índice calculado não corresponde ao termo procurado, sendo feito um novo cálculo baseado no resultado encontrado



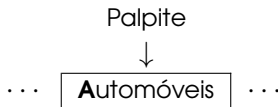
# Busca Interpolada

- ▶ Como buscar de forma interpolada?
  - ▶ Lista telefônica: automóveis
  - ▶ Por estar após o termo encontrado, é feito um novo cálculo probabilístico do índice considerando elementos da parte superior



# Busca Interpolada

- ▶ Como buscar de forma interpolada?
  - ▶ Lista telefônica: automóveis
  - ▶ O próximo índice calculado pode resultar diretamente no termo procurado ou pode ser necessária a realização de uma busca sequencial para encontrar o elemento

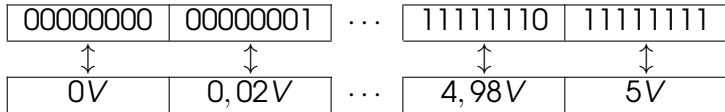


# Busca Interpolada

- ▶ Análise de complexidade
  - ▶ Índice de particionamento é calculado em tempo constante, de acordo com uma distribuição de probabilidade
  - ▶ Utilizando uma entrada com distribuição uniforme, é obtido um tempo de execução  $\Omega(1)$  com uma quantidade média de  $\log \log n$  comparações
  - ▶ No pior caso é obtida a complexidade  $O(n)$  quando uma distribuição uniforme não é aplicada

## Exemplo

- ▶ Considere o problema de quantização de dados, onde uma informação em formato analógico precisa ser representada em um formato digital
  - ▶ O sinal analógico possui amplitude de 0 a 5V
  - ▶ A codificação digital possui 8 bits para representação
  - ▶ Aplicando os conceitos de busca binária e interpolada, realize a conversão do sinal analógico 3,78V para sua representação binária de 8 bits



# Exercício

- ▶ A empresa de tecnologia Poxim Tech está realizando um estudo comparativo entre a busca binária e interpolada para um sistema de biblioteca, para determinar qual das abordagens é mais eficiente
  - ▶ Os livros são identificados unicamente pelo International Standard Book Number (ISBN) que é composto de 13 dígitos numéricos
  - ▶ Para realização de consulta dos livros é utilizado o ISBN, contabilizando o número total de chamadas realizadas para realização da busca binária e interpolada e retornando o nome do autor (até 50 caracteres) e do título do livro (até 100 caracteres)
  - ▶ A busca interpolada é feita da função de heurística 
$$h(i, j) = \lfloor i + (ISBN_j - ISBN_i) \bmod (j - i + 1) \rfloor$$
 para determinar o provável índice do livro procurado

# Exercício

## ► Formato do arquivo de entrada

- [#*Livros*]
- [#*ISBN*<sub>1</sub>] [*Autor*<sub>1</sub>] | [*Título*<sub>1</sub>]
- ...
- [#*ISBN*<sub>*n*</sub>] [*Autor*<sub>*n*</sub>] | [*Título*<sub>*n*</sub>]
- [#*Consultas*]
- [#*ISBN*<sub>1</sub>]
- ...
- [#*ISBN*<sub>*m*</sub>]

```
5
9780130224187 Niklaus Wirth | Algorithms + Data Structures = Programs
9780201416077 Gaston Gonnet | Handbook of Algorithms and Data Structures
9780262033848 Thomas Cormen | Introduction to Algorithms
9780321751041 Donald Knuth | The Art of Computer Programming
9781584884354 Dinesh Mehta | Handbook of Data Structures and Applications
3
9780130224187
9781584884354
1234567890123
```



# Exercício

- ▶ Formato do arquivo de saída
  - ▶ Para cada consulta realizada é exibida a quantidade de chamadas realizadas pela busca binária e interpolada, com informações sobre o livro
  - ▶ Após a realização das consultas é exibido a quantidade de vitórias de cada algoritmo, onde em caso de empate a busca interpolada é vencedora

[9780130224187] BINARY: 2, INTERPOLATION: 1

Author: Niklaus Wirth

Title: Algorithms + Data Structures = Programs

[9781584884354] BINARY: 3, INTERPOLATION: 3

Author: Dinesh Mehta

Title: Handbook of Data Structures and Applications

[1234567890123] BINARY: 3, INTERPOLATION: 3

ISBN NOT FOUND

SEARCH RESULTS

BINARY: 0 @ ~2 STEPS

INTERPOLATION: 3 @ ~2 STEPS