

classificação e pesquisa de dados

ordenação
comparação

1

algoritmos
pesquisa

classificação e
pesquisa
de dados

Verificar se um determinado item **encontra-se ou não** em uma dada coleção

Também denominados **algoritmos de busca**

Baseados em um **campo chave**

2

algoritmos
pesquisa

classificação e
pesquisa
de dados

Alguns tipos

Sequencial ou linear

Sequencial ordenada

Busca binária

3

algoritmos
pesquisa

classificação e
pesquisa
de dados

Algumas considerações

Como os dados estão **estruturados**?

Os dados estão **ordenados**?

Existem valores **duplicados**?

Caso sim, qual **critério** de busca?

4

pesquisa sequencial

classificação e pesquisa de dados

Também denominado: **pesquisa linear**

Percorre a coleção da primeira à última posição

Compara o conteúdo com a **chave de busca**

5

pesquisa sequencial

classificação e pesquisa de dados

	0	1	2	3	4	5	6
V	23	4	67	-8	54	90	21
	54	chave de busca					
i=0	23	4	67	-8	54	90	21
i=1	23	4	67	-8	54	90	21
i=2	23	4	67	-8	54	90	21
i=3	23	4	67	-8	54	90	21
i=4	23	4	67	-8	54	90	21



6

pesquisa sequencial

classificação e pesquisa de dados

inteiro busca_linear(v, tam, k)

Entrada: vetor v, seu respectivo tamanho e a chave k

Saída: índice onde se encontra a chave ou -1 c.c.

```
para i ← 0 até tam-1 faça
    se v[i] = k então
        retorna i
```

 $O(n)$

```
retorna -1
```

 $\Theta(1)$

$$O(n) + \Theta(1) = O(\max(n, 1)) = O(n)$$

7

pesquisa sequencial

classificação e pesquisa de dados

inteiro busca_linear(v, tam, k)

Entrada: vetor v, seu respectivo tamanho e a chave k

Saída: índice onde se encontra a chave ou -1 c.c.

```
idx ← 0
para i ← 0 até tam-1 faça
    se v[i] = k então
        idx ← i
```

 $\Theta(n)$

```
retorna idx
```

 $\Theta(1)$

$$\Theta(n) + \Theta(1) = \Theta(\max(n, 1)) = \Theta(n)$$

8

pesquisa sequencial sentinela

classificação e pesquisa de dados

inteiro busca_linear_sentinela(v, tam, k)

Entrada: vetor v, seu respectivo tamanho e a chave k

Saída: índice onde se encontra a chave ou -1 c.c.

```
ult ← v[tam-1]
```

```
i ← 0
```

```
enquanto v[i] ≠ k faça
```

```
  i ← i + 1
```

```
v[tam-1] ← ult
```

```
se i < tam-1 ou v[tam-1] = k então  
  retorna i
```

```
retorna -1
```

 $\Theta(1)$
 $O(n)$
 $\Theta(1)$
 $O(n)$

9

pesquisa sequencial sentinela

classificação e pesquisa de dados

inteiro busca_linear_sentinela(v, tam, k)

Entrada: vetor v, seu respectivo tamanho e a chave k

Saída: índice onde se encontra a chave ou -1 c.c.

```
se v[tam-1] = k então
```

```
  retorna tam-1
```

```
ult ← v[tam-1]
```

```
v[tam-1] ← k
```

```
i ← 0
```

```
enquanto v[i] ≠ k faça
```

```
  i ← i + 1
```

```
v[tam-1] ← ult
```

```
se i < tam-1 então
```

```
  retorna i
```

```
retorna -1
```

 $\Theta(1)$
 $O(n)$
 $\Theta(1)$
 $O(n)$

10

sequencial ordenada

classificação e pesquisa de dados

Possibilita **otimizar** a pesquisa

Pressupõe a **ordenação** da coleção

Compara o conteúdo com a **chave de busca**

11

sequencial ordenada

classificação e pesquisa de dados

	0	1	2	3	4	5	6
V	-8	4	21	23	54	67	90
	34 chave de busca						
i=0	-8	4	21	23	54	67	90
i=1	-8	4	21	23	54	67	90
i=2	-8	4	21	23	54	67	90
i=3	-8	4	21	23	54	67	90
i=4	-8	4	21	23	54	67	90



12

sequencial ordenada

classificação e pesquisa de dados

inteiro busca_linear_ord(v, tam, k)

Entrada: vetor v, seu respectivo tamanho e a chave k

Saída: índice onde se encontra a chave ou -1 c.c.

$i \leftarrow 0$ $\Theta(1)$

enquanto $i < \text{tam}$ e $v[i] \leq k$ **faça**
 se $v[i] = k$ **então** $\Theta(n)$
 retorna i
 $i \leftarrow i + 1$

retorna -1 $\Theta(1)$

$O(n)$

13

ordenação seleção

classificação e pesquisa de dados

Selecionar o **menor** elemento dentre as posições 0 .. n-1 do vetor e movê-lo para a primeira posição

Selecionar o **menor** elemento dentre as posições 1 .. n-1 do vetor e movê-lo para a segunda posição

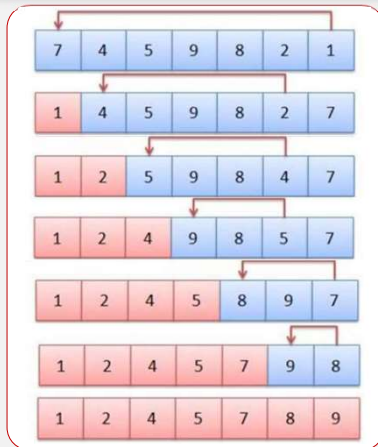
Selecionar o **menor** elemento dentre as posições 2 .. n-1 do vetor e movê-lo para a terceira posição

E, assim, sucessivamente...

14

ordenação seleção

classificação e pesquisa de dados



15

ordenação seleção

classificação e pesquisa de dados

ordenacao_selecao(v, tam)

Entrada: vetor v e seu respectivo tamanho

Saída: Não tem

para $i \leftarrow 0$ **até** tam-2 **faça**

 menor $\leftarrow i$ $\Theta(1)$

para $j \leftarrow i+1$ **até** tam-1 **faça**

se $v[j] < v[\text{menor}]$ **então**

 menor $\leftarrow j$ $\Theta(n^2)$

 aux $\leftarrow v[\text{menor}]$

 v[menor] $\leftarrow v[i]$

 v[i] \leftarrow aux $\Theta(1)$

$$\sum_{j=0}^{n-1} j = \frac{n(n-1)}{2} = \Theta(n^2)$$

$\Theta(n^2)$

$\Theta(n) \times 1 = \Theta(n)$

$\Theta(n) \times \Theta(1) = \Theta(n)$

16

ordenação inserção

classificação e pesquisa de dados

Divida o vetor em duas partes: **esq** (ordenada) e **dir** (não ord.)

Para cada elemento d de **dir**

salve o valor de d (o local de d passa a ser uma posição livre)

compare-o com cada elemento e da **esq** a partir de seu antecessor

se $d < e$ então

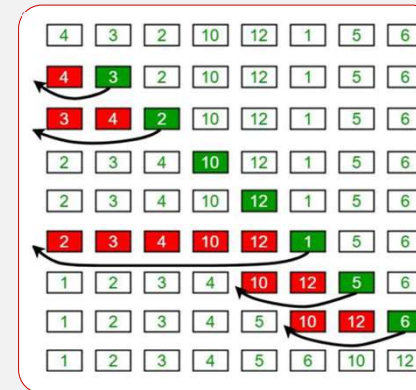
mova e uma posição para a direita (altera posição livre)

insira d na posição livre

17

ordenação inserção

classificação e pesquisa de dados



18

ordenação inserção

classificação e pesquisa de dados

`ordenacao_insercao(v, tam)`

Entrada: vetor v e seu respectivo tamanho

Saída: Não tem

para $i \leftarrow 1$ **até** $tam-1$ **faça**

$chave \leftarrow v[i]$

$j \leftarrow i - 1$

enquanto $j \geq 0$ e $v[j] > chave$ **faça**

$v[j + 1] \leftarrow v[j]$

$j \leftarrow j - 1$

$v[j + 1] \leftarrow chave$

$$\Theta(n) \times \Theta(1) = \Theta(n)$$

$$\sum_{j=1}^{n-1} \Theta(n^j) = O(n^2)$$

$$\Theta(n)$$

19

ordenação flutuação

classificação e pesquisa de dados

Passe por todos os itens do vetor, comparando-os dois a dois

Caso os itens comparados estejam invertidos, troque-os de posição

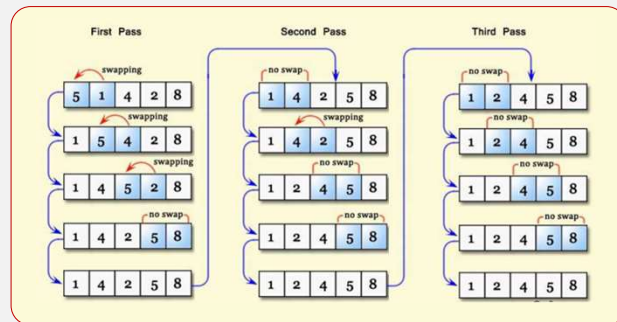
Caso contrário, avance para o próximo par de itens

Execute as instruções anteriores $(n-1)$ vezes

20

ordenação flutuação

classificação e pesquisa de dados



21

ordenação flutuação

classificação e pesquisa de dados

ordenacao_flutuacao(v, tam)

Entrada: vetor v e seu respectivo tamanho

Saída: Não tem

```

para i ← 1 até tam-1 faça
  para j ← 1 até tam-1 faça
    se v[j] < v[j-1] então
      aux ← v[j]
      v[j] ← v[j-1]
      v[j-1] ← aux
  
```

 $\Theta(n)$
 $(n-1) \times \Theta(n)$
 $\Theta((n-1) \times n)$
 $\Theta(n^2 - n)$
 $\Theta(n^2)$

22

ordenação flutuação

classificação e pesquisa de dados

ordenacao_flutuacao(v, tam)

Entrada: vetor v e seu respectivo tamanho

Saída: Não tem

```

trocou ← verdadeiro
enquanto trocou = verdadeiro faça
  trocou ← falso
  para j ← 1 até tam-1 faça
    se v[j] < v[j-1] então
      aux ← v[j]
      v[j] ← v[j-1]
      v[j-1] ← aux
      trocou ← verdadeiro
  
```

 $O(n) \times \Theta(n)$
 $O(n \times n)$
 $O(n^2)$
 $O(n^2)$

23

REFERÊNCIAS

análise: melhor e pior casos

ZIVIANI, Nivio. Projeto de Algoritmos com Implementação em Java e C++. São Paulo: Thomson Learning, 2007.

LEISERSON, C. E.; STEIN, C.; RIVEST, R. L., CORMEN, T.H. Algoritmos Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro: Campus, 2002.

LINTZMAYER, Carla Negri; MOTA, Guilherme Oliveira. Análise de Algoritmos e de Estruturas de Dados. Notas de aula. Versão: 29/06/2022.

24