

ORDENAÇÃO EM MEMÓRIA PRIMÁRIA

PARTE 1

anaeliza.moura@unicap.br

1

Métodos de Classificação em Memória Primária

■ Métodos Elementares

- Classificação por Inserção
 - Método da Inserção Direta
 - Método dos Incrementos Decrescentes - Shellsort
- Classificação por Trocas
 - Método da Bolha - Bubblesort

anaeliza.moura@unicap.br

2

Métodos de Classificação em Memória Primária

■ Métodos Elementares (cont.)

- Classificação por Seleção
 - Método da Seleção Direta
- Classificação por Intercalação
 - Método da Intercalação Simples - MergeSort

anaeliza.moura@unicap.br

3

Métodos de Classificação em Memória Primária

■ Métodos Eficientes

- Classificação por Troca
 - Método de Partição e Troca - Quicksort
- Classificação por Seleção
 - Método de Seleção em Árvore - Heapsort

anaeliza.moura@unicap.br

4

Classificação por Troca

■ Método de Troca e Partição - Quicksort

- O método de troca e partição foi proposto por Hoare.
- Comparado com os demais métodos é o que apresenta, em média, o menor tempo de classificação.
- Apresenta o menor número de operações elementares.

anaeliza.moura@unicap.br

5

Classificação por Troca

■ O Método Quicksort - Descrição

- Numa etapa inicial, o vetor a ser ordenado é particionado em três segmentos S_1 , S_2 e S_3 , da seguinte forma:
 - S_2 terá comprimento 1 e conterá uma chave denominada particionadora.
 - S_1 terá comprimento ≥ 0 e conterá todas as chaves cujos valores forem menores ou iguais ao da chave particionadora. Esse segmento é posicionado à esquerda de S_2 .
 - S_3 terá comprimento ≥ 0 e conterá todas as chaves cujos valores forem maiores do que o valor da chave particionadora. Esse segmento é posicionado à direita de S_2 .

anaeliza.moura@unicap.br

6

Classificação por Troca

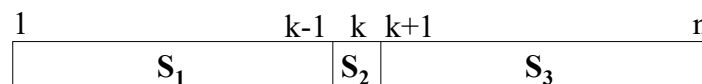
■ O Método Quicksort - Descrição

- Como $V[i] \leq V[k]$ para $i = 1, \dots, k-1$ e $V[i] > V[k]$ para $i = k+1, \dots, n$, a chave particionadora $V[k]$ ocupa sua posição definitiva correta na ordenação.

Vetor inicial



Vetor particionado



anaeliza.moura@unicap.br

7

Classificação por Troca

■ O Método Quicksort - Descrição

- O processo de particionamento é reaplicado aos segmentos S_1 e S_3 e a todos os segmentos correspondentes daí resultantes, que tiverem comprimento > 1 . Quando não restarem mais segmentos a serem particionados, o vetor estará ordenado.

anaeliza.moura@unicap.br

8

Classificação por Troca

■ Método Quicksort - Escolha da Chave Particionadora

- A chave particionadora ideal é aquela que produz segmentos S_1 e S_3 de tamanho igual (ou aproximado).
- Para isso, a chave particionadora deve ser a chave de valor mediano.
- Porém, seria necessário, para cada iteração, percorrer todo o vetor para encontrar tal chave, o que seria ineficiente.

anaeliza.moura@unicap.br

9

Classificação por Troca

■ Método Quicksort - Escolha da Chave Particionadora

- Para evitar esse problema, devemos utilizar um critério de escolha simples e rápido.
- Como não temos conhecimento prévio sobre a distribuição dos valores das chaves, qualquer uma delas pode ser a particionadora.
- Por questões de simplicidade, vamos escolher a chave que se encontra na posição inicial do vetor para ser a particionadora.

anaeliza.moura@unicap.br

10

Classificação por Troca

■ Método Quicksort - Algoritmo

- O algoritmo executa o particionamento em **N** passos (**N** = número de chaves). Nos **N-1** primeiros, as chaves (excluída a particionadora) são deslocadas para o lado esquerdo, se menores ou iguais a particionadora, ou para o lado direito, se maiores. No último passo, a chave particionadora é inserida na sua posição correta.

anaeliza.moura@unicap.br

11

Classificação por Troca

■ Método Quicksort - Exemplo

Vetor inicial: 9 25 10 18 5 7 15 3

Escolhemos a chave 9 como particionadora e a guardamos em uma variável temporária (CP). Com isso a posição ocupada por ela se torna disponível para ser ocupada por outra chave. Esta situação é indicada pelo símbolo □:

□ 25 10 18 5 7 15 3 CP = 9

anaeliza.moura@unicap.br

12

Classificação por Troca

■ Método Quicksort - Exemplo (cont.)

Passo 1: Marcamos o início e o fim do vetor por dois ponteiros: **i** (de início) e **f** (de fim). A expressão “esquerda” escrita ao lado do vetor indica que a posição apontada pelo ponteiro **i** (a da esquerda) está disponível e pode ser ocupada por outra chave.

□	25	10	18	5	7	15	3	CP = 9	“esquerda”
↑							↑		
i							f		

anaeliza.moura@unicap.br

13

Classificação por Troca

■ Método Quicksort - Exemplo (cont.)

Passo 2: Comparamos a chave que está apontada por **f** com a particionadora. Como 3 é menor do que 9, a deslocamos para o lado esquerdo do vetor, ao mesmo tempo que avançamos o ponteiro **i** para indicar que a chave recém-movida já está no segmento certo. A nova posição vaga passa a ser a apontada por **f**, ou seja, a da “direita”:

3	25	10	18	5	7	15	□	CP = 9	“direita”
↑							↑		
i							f		

anaeliza.moura@unicap.br

14

Classificação por Troca

■ Método Quicksort - Exemplo (cont.)

Passo 3: Agora comparamos a chave 25 (pois a nova posição vaga é a da direita) com a particionadora. Como 25 é maior, a deslocamos para a posição vaga, ao mesmo tempo que recuamos o ponteiro f para indicar que a chave recém-movida já está no segmento certo. A nova posição vaga passa a ser a apontada por i, ou seja, a da “esquerda”:

3	□	10	18	5	7	15	25	CP = 9	“ <i>esquerda</i> ”
	↑					↑			
	i					f			

anaeliza.moura@unicap.br

15

Classificação por Troca

■ Método Quicksort - Exemplo (cont.)

Passo 4: O processo prossegue comparando a chave 15. Neste caso, por ela ser maior do que a particionadora, não deve ser trocada de posição, pois já se encontra no segmento correto. Apenas o ponteiro f é deslocado para a esquerda:

3	□	10	18	5	7	15	25	CP = 9	“ <i>esquerda</i> ”
	↑					↑			
	i					f			

anaeliza.moura@unicap.br

16

Classificação por Troca

■ Método Quicksort - Exemplo (cont.)

Passo 5:

3 7 10 18 5 □ 15 25 CP = 9 “direita”
 ↑ ↑
 i f

Passo 6:

3 7 □ 18 5 10 15 25 CP = 9 “esquerda”
 ↑ ↑
 i f

anaeliza.moura@unicap.br

17

Classificação por Troca

■ Método Quicksort - Exemplo (cont.)

Passo 7:

3 7 5 18 □ 10 15 25 CP = 9 “direita”
 ↑ ↑
 i f

- O resultado do passo 7 (passo $n - 1$) produz a situação abaixo, na qual os dois ponteiros se encontram. Quando isso ocorre, os segmentos S_1 e S_3 já estão formados. A posição vaga é a posição do segmento S_2 .

3 7 5 □ 18 10 15 25 CP = 9
 ↑
 i,f

anaeliza.moura@unicap.br

18

Classificação por Troca

■ Método Quicksort - Exemplo (cont.)

Passo 8: Resta copiar o valor da chave particionadora na posição apontada por i e j e o processo de particionamento estará completado:

Vetor final: 3 7 5 9 18 10 15 25

┌───┬───┬───┐
S1 S2 S3

- Embora os segmentos S_1 e S_3 ainda não estejam ordenado, a chave particionadora já se encontra em sua posição correta definitiva.
- Reaplicar o método recursivamente para os segmentos S_1 e S_3 até que o vetor esteja completamente ordenado.

anaeliza.moura@unicap.br

19

Classificação por Troca

■ Método Quicksort - Implementação

```
public static void quicksort ( <tipo_componente> v [ ], int i, int f) {  
    int k;  
    if (f > i) { // tamanho do segmento é maior que 1?  
        k = partição (v, i, f);  
        quicksort (v,i,k-1);    // ordenar segmento S1  
        quicksort (v, k+1, f);  // ordenar segmento S3  
    }  
}
```

anaeliza.moura@unicap.br

20