

Classificação e Pesquisa de Dados

Aula 05
Classificação de dados por Troca : Bubblesort

UFRGS

INF01124

1

Classificação por Trocas

Classificação por comparação entre pares de chaves, trocando-as de posição caso estejam fora de ordem no par

Principais Algoritmos

- Bubblesort
- Quicksort

2

Bubblesort Método da Bolha

- ◆ Compara todos os pares consecutivos (adjacentes no vetor) de chaves, realizando troca caso necessário
- ◆ Realiza um certo número de varreduras (passadas) sobre o vetor a ser ordenado
- ◆ O procedimento termina quando, em uma dada varredura, nenhuma troca de chaves ocorre

3

Bubblesort Exemplo

Suponha que se deseja classificar em ordem crescente o seguinte vetor de chaves [28, 26, 30, 24, 25]

Primeira Varredura

28	26	30	24	25	compara par (28, 26) : troca
26	28	30	24	25	compara par (28, 30) : não troca
26	28	30	24	25	compara par (30, 24) : troca
26	28	24	30	25	compara par (30, 25) : troca
26	28	24	25	30	fim da primeira varredura

Maior chave se encontra em sua posição definitiva

4

Bubblesort Exemplo (Cont.)

Segunda Varredura

26	28	24	25	30	compara par (26, 28) : não troca
26	28	24	25	30	compara par (28, 24) : troca
26	24	28	25	30	compara par (28, 25) : troca
26	24	25	28	30	(não precisa comparar) fim da segunda varredura

5

Bubblesort Exemplo (Cont.)

Terceira Varredura

26	24	25	28	30	compara par (26, 24) : troca
24	26	25	28	30	compara par (26, 25) : troca
24	25	26	28	30	fim da terceira varredura

Durante a quarta varredura, nenhuma troca ocorrerá e a execução do algoritmo terminará

6

Bubblesort Exercícios

Considerando o seguinte vetor :

13	11	25	10	18	21	23
----	----	----	----	----	----	----

- 1) Realize a ordenação do vetor utilizando o bubblesort.
- 2) Em quantas varreduras o vetor é classificado ?
- 3) Como identificar, após uma dada varredura, quantas chaves já estão classificadas?

No caso de ordenação não decrescente, em cada varredura o vetor encontrar-se-á ordenado a partir da posição onde ocorreu a última troca!

7

Procedimento Bubblesort

```

Proc bubblesort ( C , n );
begin /* C: vetor a ser ordenado, n é o número de elementos no vetor */
  troca ← true, m ← n - 1; k ← 1;
  while troca do
    begin
      troca ← false;
      for i ← 1 to m do
        if C[i] > C[i + 1]
          then begin /* troca */
              ch ← C[i];
              C[i] ← C[i + 1];
              C[i + 1] ← ch;
              k ← i; /* posição da última troca */
              troca ← true /* sinaliza ocorrência de troca */
            end
      end
      m ← k /* vetor já ordenado de m + 1 até n */
    end
  end
end

```

8

Bubblesort Análise de Desempenho

◆ Melhor caso

- ◆ Quando o vetor já se encontra ordenado
- ◆ Nenhuma troca ocorre ainda na primeira varredura
- ◆ Custo linear: n-1 comparações

◆ Pior Caso

- ◆ Quando o vetor se encontra na ordem inversa a desejada
- ◆ A cada varredura apenas uma chave será colocada em sua posição definitiva

9

Bubblesort Análise de Desempenho

◆ Pior Caso (Cont.)

Nro. varredura	Comparações efetuadas
1	n - 1
2	n - 2
3	n - 3
...	...
n - 1	1
Total	$(n^2 - n)/2$

10

Exercícios

- ◆ Expresse os limites superior e inferior do algoritmo utilizando as notações O e Ω .
- ◆ É possível utilizar a notação Θ para expressar o custo assintótico do bubblesort?
- ◆ Classifique o algoritmo bubblesort como:
 - ◆ *Estável versus não-estável*
 - ◆ *Local versus não-local*

11