

Classificação e Pesquisa de Dados

Aula 10
Classificação por Intercalação: MergeSort

UFRGS

INF01124

1

Classificação por Intercalação Merge Sort

- ◆ Utiliza o paradigma de Dividir para Conquistar
 - **Dividir:** Divide a sequência de n elementos a ser classificada em duas subseqüências de $n/2$ elementos
 - **Conquistar:** Classifica as duas subseqüências recursivamente utilizando o merge sort
 - **Combinar:** Intercala as subseqüências ordenadas
- A operação chave é a intercalação de duas seqüências ordenadas
 - O processo de intercalação inicia com segmentos de comprimento 1, e prossegue até que toda a seqüência esteja classificada

2

Classificação por Intercalação Merge Sort

Exemplo:

23	17	8	15	9	12	19	7
17	23	8	15	9	12	7	19
8	15	17	23	7	9	12	19
7	8	9	12	15	17	19	23

3

Procedimento Merge Sort

```
Proc merge-sort (A, i, f)
/* i: posição inicial do vetor;
   f: posição final do vetor */

begin
  if (i < f) then
    begin
      m ← ⌊(i + f)/2⌋
      merge-sort (A, i, m);
      merge-sort (A, m+1, f);
      merge(A, i, m, f); /* intercala as duas seqüências ordenadas */
    end
  end
end
```

4

Exercícios

- ◆ Escrever um procedimento para fazer a intercalação de duas seqüências ordenadas
- ◆ Qual o custo deste procedimento?

5

Merge Sort Análise de Desempenho

$$T(n) = \underbrace{2T(n/2)}_{\text{Tempo para ordenar toda a seqüência}} + \underbrace{D(n)}_{\text{Tempo para ordenar duas seqüências de tamanho } n/2} + \underbrace{C(n)}_{\text{Tempo gasto para dividir seqüência}} + \underbrace{C(n)}_{\text{Tempo combinar os resultados - intercalação}}$$

```
Proc merge-sort (A, i, f)
begin
  if (i < f) then
    begin
      m ← ⌊(i + f)/2⌋
      merge-sort (A, i, m);
      merge-sort (A, m+1, f);
      merge(A, i, m, f);
    end
  end
end
```

Profundidade da árvore de recursão é $\log_2 n$
Consequentemente, o algoritmo apresenta custo $\Theta(n \log_2 n)$

6