

Exercício 2:

Reescreva a rotina *bubble* com sucessivas passagens em direções opostas.

Exercício 3:

Ilustre a operação do *quicksort* sobre o vetor  $x = \langle 5, 13, 2, 25, 7, 17, 20, 8 \rangle$ .

## Exercício 4:

Considere a seguinte classificação por seleção quadrática: divida os  $n$  elementos do vetor em  $n$  raiz quadrado grupos de  $n$  raiz quadrado elementos cada. Encontre o maior elementos de cada grupo e insira-o num vetor auxiliar. Esse será o maior elementos do vetor. Em seguida, substitua esse elemento dentro do vetor pelo maior elemento seguinte do grupo a que ele pertencia. Ache novamente o maior elemento do vetor auxiliar. Esse será o segundo maior elemento do vetor. Repita o processo até que o arquivo esteja classificado. Escreve um algoritmo de classificação por seleção quadrática o mais eficiente possível.

Exercício 5:

A seqüência  $\langle 23, 17, 14, 6, 13, 10, 1, 5, 7, 12 \rangle$  é um heap?

Exercício 6:

Ilustre a operação de Heapsort sobre o vetor  $A = \langle 5, 13, 2, 25, 7, 17, 20, 8, 4 \rangle$

## Exercício 7:

Defina uma árvore ternária quase completa como uma árvore na qual todo nó tem, no máximo, três filhos, e na qual os nós podem ser numerados de 0 a  $n-1$ , de modo que os filhos de  $\text{node}[i]$  seja  $\text{node}[3*i+1]$ ,  $\text{node}[3*i+2]$  e  $\text{node}[3*i+3]$ . Defina um heap ternário como uma árvore ternária quase completa na qual o conteúdo de cada nó é maior ou igual ao conteúdo de todos seus filhos. Escreva uma rotina de classificação semelhante ao heapsort usando um heap ternário.

Exercício 8:

Desenvolva um algoritmo usando um heap de  $k$  elementos para achar os maiores  $k$  números num grande vetor não-ordenado, de  $n$  números.