

Exercícios Ordenação

- 1) Ilustre uma partição da execução do quicksort no vetor $A = [13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 21]$ considerando os pivôs $= A[n]$ e $A[n/2]$.
- 2) Construa uma heap máxima sobre o vetor $A = [13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 21]$.
- 3) Mostre que qualquer algoritmo de ordenação por comparação possui limite inferior $n \log n$. Dica: usar árvore de decisão.
- 4) É possível tornar qualquer algoritmo de ordenação estável? Quanto tempo ou memória adicional seu esquema requer?
- 5) Ilustre a ordenação por contagem sobre o vetor $A = [6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2]$.
- 6) Se alterarmos o **loop for** na linha 9 para: **for j <-1 to comprimento[A]** o algoritmo ainda funciona corretamente? Ele ainda é estável?

```
COUNTING-SORT( $A, B, k$ )
1  for  $i \leftarrow 0$  to  $k$ 
2    do  $C[i] \leftarrow 0$ 
3  for  $j \leftarrow 1$  to comprimento[A]
4    do  $C[A[j]] \leftarrow C[A[j]] + 1$ 
5  ▷ Agora  $C[i]$  contém o número de elementos iguais a  $i$ .
6  for  $i \leftarrow 2$  to  $k$ 
7    do  $C[i] \leftarrow C[i] + C[i - 1]$ 
8  ▷ Agora  $C[i]$  contém o número de elementos menores que ou iguais a  $i$ .
9  for  $j \leftarrow comprimento[A]$  downto 1
10   do  $B[C[A[j]]] \leftarrow A[j]$ 
11    $C[A[j]] \leftarrow C[A[j]] - 1$ 
```

- 7) Ilustre a ordenação radix-sort sobre a seguinte lista de palavras: cow, dog, sea, rug, row, mob, box, tab, bar, ear, tar, dig, big, tea, now, fox.