ARA7125 Estruturas de Dados I
Primeiro Semestre - 2015
Quarta Lista de Exercícios

Considere o seguinte algoritmo que ordena em ordem crescente um vetor de inteiros $A[1..n]$ com n elementos:

ORDENAÇÃO_POR_INSERÇÃO(A, n)

```
1  para j ← 2 até n
2      faça chave ← A[j]
3           i ← j − 1
4           enquanto i ≥ 1 e A[i] > chave
5               faça A[i + 1] ← A[i]
6                    i ← i − 1
7           A[i + 1] ← chave
```

1. (CLRS) – Reescreva o algoritmo `Ordenação_por_Inserção` para ordenar em ordem decrescente.

2. (PF) – Escreva uma versão recursiva para `Ordenação_por_Inserção`.

3. (PF) – Note que as linhas do comando `enquanto` do algoritmo `Ordenação_por_Inserção` (linhas 4 – 6) têm o objetivo de encontrar o ponto onde `chave` (que é igual a `A[j]` (veja linha 2)) deve ser inserido no subvetor ordenado `A[1..j-1]`. Escreva uma nova versão para este algoritmo de ordenação que usa um algoritmo de busca binária neste ponto. Neste caso, o uso da busca binária deixa a ordenação mais rápida?

4. – Escreva uma versão recursiva para o algoritmo `Seleção` (ordenação por seleção visto em sala de aula).

5. – ("Problema da ordenação indireta".) Escreva um algoritmo que recebe um número inteiro n e um vetor constante de números inteiros $A[1..n]$ e devolve um vetor $B[1..n]$ de índices do vetor A de tal forma que $A[B[1]] \leq A[B[2]] \leq \ldots \leq A[B[n]]$. Os valores armazenados em um vetor constante não podem ser alterados. Se A é um vetor constante e $A[1] = 5$, então $A[1] = 5$ para sempre.

Ex.: Suponha uma entrada $A = [3, 6, 1, 8]$. O seu algoritmo deve devolver $B = [3, 1, 2, 4]$, pois, $A[B[1]] = A[3] = 1 \leq A[B[2]] = A[1] = 3 \leq A[B[3]] = A[2] = 6 \leq A[B[4]] = A[4] = 8$.

6. – O que é um `max-heap`? Onde um `max-heap` pode ser útil?

7. – Escreva uma nova versão do algoritmo `QuickSort` que não aplica recursão a vetores com 2 elementos ou menos.

8. – Descreva detalhadamente o processo de ordenação do vetor a seguir usando os algoritmos `MergeSort` e `HeapSort` (incluíndo a construção do `max-heap`) e `QuickSort`.

$$A = [79, 101, 3, 6, 13, 7, 34, 35, 18].$$

9. – Implemente os algoritmos de ordenação (Inserção, Seleção, MergeSort, QuickSort, e HeapSort) para ordenar (em ordem crescente) um vetor de caracteres. Por exemplo, para o vetor $A = [b, a, n, a, n, a]$, o resultado deve ser $A = [a, a, a, b, n, n]$.

10. – Implemente os algoritmos de ordenação (Inserção, Seleção, MergeSort, QuickSort, e HeapSort) para ordenar (em ordem crescente) um vetor de números reais. Por exemplo, para o vetor $A = [0.000000001, 0.00000000099]$, o resultado deve ser $A = [0.00000000099, 0.000000001]$.