

1. (1.0 ponto) Resolva o exercício a seguir:

Exercício 84. Prove que o algoritmo BubbleSort a seguir é correto.

```
1 for i = 0 to n - 2 do
2   for j = 0 to n - 2 - i do
3     if A[j + 1] < A[j] then
4       swap A[j] and A[j + 1];
5     end
6   end
7 end
```

Algorithm 3: BubbleSort($A[0..n-1]$)

Invariante do laço interno (linhas 2-6): (0.5 ponto)

Antes da $(j+1)$ -ésima iteração do laço interno, o maior elemento do subvetor $A[0..j]$ está na posição j .

Inicialização: Antes da primeira iteração, temos $j=0$, e o maior elemento do subvetor $A[0]$ está na posição 0 (trivial).

Manutenção: Suponha que antes da k -ésima iteração ($1 < k \leq n-1-i$), o maior elemento do subvetor $A[0..k-1]$ esteja na posição $k-1$. Durante a k -ésima iteração, temos $j=k-1$, e portanto os elementos $A[j]=A[k-1]$ e $A[j+1]=A[k]$ serão comparados. Se $A[k]=A[j+1] < A[j]=A[k-1]$ então $A[k-1]$ é o maior dos elementos do subvetor $A[0..k]$ que é colocado na posição k após a permuta feita na linha 4. Caso contrário, i.e. se $A[k]=A[j+1] \geq A[j]=A[k-1]$ então $A[k]$ é o maior dos elementos do subvetor $A[0..k]$ que permanece na posição k , preservando assim a invariante.

Terminação: Ao final do laço, temos $j=n-2-i+1=n-1-i$, e a invariante nos diz que o maior elemento do subvetor $A[0..n-1-i]$ está na posição $n-1-i$.

Invariante do laço externo (linhas 1-7) (0.5 ponto)

Antes da $(i+1)$ -ésima iteração, os i -ésimos maiores elementos do vetor A estão ordenados no subvetor $A[n-i..n-1]$.

Inicialização: Antes da primeira iteração, i.e. $i=0$, temos que os 0-ésimos maiores elementos do vetor A estão ordenados no subvetor vazio $A[n..n-1]$. Logo, a invariante está satisfeita por vacuidade.

Manutenção: Suponha que antes da k -ésima iteração ($1 < k \leq n-1$) os $(k-1)$ -ésimos maiores elementos do vetor A estejam ordenados

no subvetor $A[n-k+1..n-1]$. Durante a k -ésima iteração pela invariante do laço interno, temos que o maior elemento do subvetor $A[0..n-1-(k-1)] = A[0..n-k]$ está na posição $n-k$. Como $A[n-k] \leq A[n-k+1]$, pois caso contrário nessa suposição seria falsa, concluímos que ao final da k -ésima iteração o subvetor $A[n-k..n-1]$ está ordenado com os k maiores elementos do vetor A preservando a invariante.

Terminação: Ao final da execução do laço externo, temos $i = n-1$ e a invariante nos diz que o subvetor $A[n-i..n-1] = A[n-(n-1)..n-1] = A[1..n-1]$ está ordenado e possui os $(n-1)$ -ésimos maiores elementos do vetor A . Então $A[0]$ é menor ou igual a todos os elementos de A , e portanto ao final da execução de bubblesort o vetor A estará ordenado com os mesmos elementos do vetor original. Assim, concluímos que bubble sort é correto. \square

2. (0.5 ponto) Faça a análise assintótica do algoritmo bubblesort acima.

Contaremos o número de comparações (operações básicas) realizadas na linha 4:

$$T(n) = \sum_{i=0}^{n-2} \left(\sum_{j=0}^{n-2-i} 1 \right) = \sum_{i=0}^{n-2} (n-2-i+1) = \sum_{i=0}^{n-2} (n-i-1) = \sum_{i=1}^{n-1} i = \frac{(n-1) \cdot n}{2} = \Theta(n^2).$$

Observe que qualquer que seja o vetor a ser ordenado, o número de comparações será sempre a mesma função de n . Ou seja, para ordenar qualquer vetor contendo n elementos, o algoritmo acima fará $\frac{(n-1) \cdot n}{2}$ comparações.

Isto significa que a análise do pior caso $T_w(n)$ e melhor caso, $T_b(n)$ coincidem neste caso:

$$T_b(n) = T_w(n) = \frac{(n-1) \cdot n}{2} = \Theta(n^2). \quad \square$$

3. (0.5 ponto) Resolva o exercício a seguir:

Mostre que $n^3 \neq O(n^2)$.

Vamos utilizar o raciocínio de redução ao absurdo, i.e. vamos supor que $n^3 = O(n^2)$ e derivar uma contradição.

Se $n^3 = O(n^2)$ então existem constantes positivas c e n_0 tais que

$$n^3 \leq c \cdot n^2, \forall n \geq n_0. \Leftrightarrow$$

$$n \leq c, \forall n \geq n_0. \quad \Leftarrow \quad \square$$