

MO417 - ATA de Exercício

Priscila Tiemi Maeda Saito - RA100576

08 de março de 2010

Exercício 2.2-2

Considere a ordenação de n números no arranjo A , localizando primeiro o menor elemento de A e permutando esse elemento com o elemento contido em $A[1]$. Em seguida, encontre o segundo menor elemento de A e o troque pelo elemento $A[2]$. Continue dessa maneira para os primeiros $n - 1$ elementos de A . Escreva o pseudocódigo para esse algoritmo, conhecido como ordenação por seleção. Que *loop* invariante esse algoritmo mantém? Porque ele só precisa ser executado para os primeiros $n - 1$ elementos, e não para todos os n elementos? Forneça os tempos de execução do melhor caso e do pior caso da ordenação por seleção em notação Θ .

Algoritmo 1: Selection-Sort(A)

```
1 begin
2   for  $i \leftarrow 1$  to  $n - 1$  do
3     menor  $\leftarrow i$ ;
4     for  $j \leftarrow i + 1$  to  $n$  do
5       se  $A[j] < A[\text{menor}]$  então
6         | menor  $\leftarrow j$ ;
7       fim
8     end
9     aux  $\leftarrow A[i]$ ;
10     $A[i] \leftarrow A[\text{menor}]$ ;
11     $A[\text{menor}] \leftarrow aux$ ;
12  end
13 end
```

O algoritmo mantém o *loop* invariante:

$A[1..i - 1]$ está ordenado e $A[i - 1] \leq A[i..n]$

(Todos os elementos já analisados até então, $A[i - 1]$, encontram-se ordenados e os elementos já ordenados são menores que os demais elementos, $A[i..n]$).

Ele só precisa ser executado para os primeiros $(n - 1)$ elementos e não para todos os n elementos, pois o último elemento do arranjo A já será o maior e, portanto, não precisa ser ordenado.

Os tempos de execução tanto no melhor quanto no pior caso são $\Theta(n^2)$.