GabaritoExercicios_3

2.2-2 Considere a ordenação de n números armazenados no arranjo A, localizando primeiro o menor elemento de A e permutando esse elemento com o elemento contido em $A[1]$. Em seguida, determine o segundo menor elemento de A e permute-o com $A[2]$. Continue dessa maneira para os primeiros $n - 1$ elementos de A. Escreva o pseudocódigo para esse algoritmo, conhecido como **ordenação por seleção**. Qual invariante de laço esse algoritmo mantém? Por que 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 OrdenaçãoPorSeleção(A,n)
    for i = 1 to n-1
        iaux = i
        for j = i+1 to n
            if(A[j] < A[iaux])
                iaux = j
        //troca A[i] por A[iaux]
        aux = A[i]
        A[i] = A[iaux]
        A[iaux] = aux
```

Invariante de laço: O subVetor A[1...i-1] consiste de elementos ordenados em ordem crescente

Inicialização: Antes do loop do for externo o subVetor[1...i-1] só tem o elemento A[1], portanto está trivialmente ordenado

Manutenção: Em cada laço do for externo, os elementos do subVetor[1... i-1] estão ordenados

Término: Ao sair do laço externo do for como A[1... n] estão ordenados, obviamente que A[1...i-1] também estrão ordenados.

Por que ele só precisa ser executado para os primeiros $n - 1$ elementos, e não para todos os n elementos?

Após n-1 interações o subVetor A[1...n-1] consiste dos menores i-1 elementos do vetor A, portanto o elemento A[n] é o maior elemento do vetor.

Forneça os tempos de execução do melhor caso e do pior caso da ordenação por seleção em notação Θ.

Algoritmo OrdenaçãoPorSeleção(A,n)	custo	vezes
for i = 1 to n-1	c_1	n
iaux = i	c_2	n-1
for j = i+1 to n	c_3	$\sum_2^n t_j$
if(A[j] < A[iaux])	c_4	$\sum_2^n t_j -1$
iaux = j	c_5	$\sum_2^n t_j -1$
//troca A[i] por A[iaux]		
aux = A[i]	c_6	n-1
A[i] = A[iaux]	c_7	n-1
A[iaux] = aux	c_8	n-1

$T(n) = c_1 n + c_2(n-1) + c_3(n(n+1)/2 -1) + c_4(n(n-1)/2) + c_5(n(n-1)/2) + c_6(n-1) + c_7(n-1) + c8(n-1)$

1	2	3	4	5	6
5	2	4	6	1	3

Primeiro Loop: para o for externo iaux =1

Para o for interno A[2] < A[1] -> iaux = 2

A[3] > A[2]

A[4] > A[2]

A[5] < A[2] -> iauux = 5

A[6] > A[5]

1	2	4	6	5	3

Segundo Loop: para o for externo iaux = 2

Para o for interno A[3] > A[2]

A[4] > A[2]

A[5] > A[2]

1 2 3 4 5 6 A[6] > A[2]

1	2	4	6	5	3

Terceiro Loop: para o for externo iaux = 3

Para o for interno A[4] > A[3]

Para o for interno A[5] > A[3]

Para o for interno A[6] < A[3] -> iaux = 6

1 2 3 4 5 6

1	2	3	6	5	4

Quarto Loop: Para o for externo iaux = 4

Para o for interno A[5] < A[4] -> iaux = 5

Para o for interno A[6] < A[5] -> iaux = 6

1	2	3	4	5	6

Quinto Loop: Para o for externo iaux = 5

Para o for interno A[6] > A[5]

1	2	3	4	5	6

Para o melhor caso $c_5 = 0$ Logo

$T(n) = c_1n + c_2(n-1) + + c_3(n(n+1)/2 -1) + c_4(n(n-1)/2) + c_6(n-1) + c_7(n-1) + c8(n-1)$

Portanto $\Theta(n^2)$

Para o pior caso também $\Theta(n^2)$

2.2 -3 Considere mais uma vez a busca linear (veja Exercício 2.1-3). Quantos elementos da sequência de entrada precisam ser verificados em média, considerando que o elemento que está sendo procurado tenha a mesma probabilidade de ser qualquer elemento no arranjo? E no pior caso? Quais são os tempos de execução do caso médio e do pior caso da busca linear em notação Θ? Justifique suas respostas.

Algoritmo BuscaLinear (A, v)		custo	vezes
1	para i = 1 até A,comprimento	c_1	n+1
2	se v == A[i]	c_2	n
3	return i	c_3	1
4	return NIL	c_4	1

$T(n) = c_1(n+1) +c_2n + c_3 + c_4$

Metade dos elementos devem ser verificados em média.

Pior caso $\Theta(n)$, mas $T(n) = c_1(n+1) + c_2n + = (c_1 + c_2)n + c_3 + c_4$

Melhor caso $\Theta(3)$

Caso médio $\Theta(n)$, mas $T(n) = c_1(n+1)/2 + c_2n/2 + c_3 + c_4= (c_1 +1+ c_2)n/2 +c_3 + c_4$

2.2-4 Como podemos modificar praticamente qualquer algoritmo para ter um bom tempo de execução no melhor caso?

Você pode modificar qualquer algoritmo para ter um bom trempo de execução de melhor caso, adicionando um caso especial. Se a entrada corresponder a este caso especial, retorne como resposta o valor pré-calculada.