

## EXERCÍCIOS

1- void maiorMenor (int array[]) {

Int maior = array[0];

Int menor = array[0];

For(int i= 1; i< n; i++){

If(array[i] < menor

menor = array[i];

If(array[i] > maior

maior = array[i];

}

}

Função de complexidade:  $f(n) = 2(n - 1)$ .

3-

|  |   |   |   |  |   |   |  |
|--|---|---|---|--|---|---|--|
|  | X |   |   |  |   |   |  |
|  |   |   | X |  |   |   |  |
|  |   | X |   |  |   |   |  |
|  |   |   |   |  |   | X |  |
|  |   |   |   |  | X |   |  |
|  |   |   |   |  |   | X |  |

4-

|  |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|
|  | X | X | X | X | X | X | X |
|  |   |   | X | X | X | X | X |
|  |   | X | X | X | X | X | X |
|  |   |   |   |   |   | X | X |
|  |   |   |   |   | X | X | X |
|  |   |   |   |   |   | X | X |

5-

|   |   |   |   |   |   |   |  |
|---|---|---|---|---|---|---|--|
| X | X |   |   |   |   |   |  |
| X | X | X | X |   |   |   |  |
| X | X | X |   |   |   |   |  |
| X | X | X | X | X | X | X |  |
| X | X | X | X | X | X |   |  |
| X | X | X | X | X | X | X |  |

6- a)  $c = 3$  e  $m = 1$

b)  $c = 3$  e  $m = 1$

c) Não existe par  $(c, m)$  para  $n \geq m$ ,  $|3n^2 + 5n + 1| \geq c \times |n \cdot 3|$  seja verdadeira.

7- a)  $c_1 = 3$ ,  $c_2 = 4$  e  $m = 5.2$

b) Não existe par  $(c_2, m)$  tal que para  $n \geq m$ ,  $|3n^2 + 5n + 1| \leq c_2 \times |n|$  seja verdadeira.

c) Não existe par  $(c_1, m)$  tal que para  $n \geq m$ ,  $|3n^2 + 5n + 1| \geq c_1 \times |n^3|$  seja verdadeira.

8- a) Melhor caso: função  $f(n) = 1$  e complexidade 1.

Pior caso: função  $f(n) = 1 + (n-2)$  e complexidade  $n$ .

b) Melhor caso: função  $f(n) = 2 + (n-2)$  e complexidade  $n$ .

Pior caso: função  $f(n) = 2 + 2 \times (n-2)$  e complexidade  $n$ .

9- 

```
for(int i= 0; i< n; i++)
```

```
{
```

```
    for(int j= 2; j< n; j++)
```

```
    {
```

```
        contador++;
```

```
    }
```

```
    if(array[i] > 10)
```

```
    {
```

```
        contador2++;
```

```
        contador3++;
```

```
    }
```

```
}
```

Melhor caso=> função:  $f(n) = n \times (n-2)$ ; complexidade:  $n^2$ .

Pior caso=> função:  $f(n) = 2 \times n \times ((n-2))$ ; complexidade:  $n^2$ .

10- A solução de ordenar e realizar pesquisas binárias é mais eficiente, pois ao realizar  $n$  pesquisas, o custo da pesquisa sequencial será de  $n * n = n^2$ ; já o da ordenação+pesquisa binária será de

$(n * \lg(n)) + n * (\lg(n)) = n * \lg(n)$ .  $n^2$  tem custo maior que  $n * \lg(n)$ .