

Ciências da Computação – PUC Minas
Algoritmo e Estrutura de Dados II
Trabalho Teórico 6

```
1) for (int i = 0; i < n; i++) {
    if(maior < array[i]){
        maior = array[i];
    }
    if (menor > array[i]) {
        menor = array[i];
    }
}
```

2)

	$O(1)$	$O(\lg n)$	$O(n)$	$O(n \cdot \lg(n))$	$O(n^2)$	$O(n^3)$	$O(n^5)$	$O(n^{20})$
$f(n) = \lg(n)$		✓	✓	✓	✓	✓	✓	✓
$f(n) = n \cdot \lg(n)$				✓	✓	✓	✓	✓
$f(n) = 5n + 1$			✓	✓	✓	✓	✓	✓
$f(n) = 7n^5 - 3n^2$							✓	✓
$f(n) = 99n^3 - 1000n^2$						✓	✓	✓
$f(n) = n^5 - 99999n^4$							✓	✓

3)

	$\Omega(1)$	$\Omega(\lg n)$	$\Omega(n)$	$\Omega(n \cdot \lg(n))$	$\Omega(n^2)$	$\Omega(n^3)$	$\Omega(n^5)$	$\Omega(n^{20})$
$f(n) = \lg(n)$	✓	✓						
$f(n) = n \cdot \lg(n)$	✓	✓	✓	✓				
$f(n) = 5n + 1$	✓	✓	✓					
$f(n) = 7n^5 - 3n^2$	✓	✓	✓	✓	✓	✓	✓	
$f(n) = 99n^3 - 1000n^2$	✓	✓	✓	✓	✓	✓		
$f(n) = n^5 - 99999n^4$	✓	✓	✓	✓	✓	✓	✓	

4)

	$\Omega(1)$	$\Omega(\lg n)$	$\Omega(n)$	$\Omega(n \cdot \lg(n))$	$\Omega(n^2)$	$\Omega(n^3)$	$\Omega(n^5)$	$\Omega(n^{20})$
$f(n) = \lg(n)$	✓	✓						
$f(n) = n \cdot \lg(n)$	✓	✓	✓	✓				
$f(n) = 5n + 1$	✓	✓	✓					
$f(n) = 7n^5 - 3n^2$	✓	✓	✓	✓	✓	✓	✓	
$f(n) = 99n^3 - 1000n^2$	✓	✓	✓	✓	✓	✓		
$f(n) = n^5 - 99999n^4$	✓	✓	✓	✓	✓	✓	✓	

5)

	$\Theta(1)$	$\Theta(\lg n)$	$\Theta(n)$	$\Theta(n \cdot \lg(n))$	$\Theta(n^2)$	$\Theta(n^3)$	$\Theta(n^5)$	$\Theta(n^{20})$
$f(n) = \lg(n)$		✓						
$f(n) = n \cdot \lg(n)$				✓				
$f(n) = 5n + 1$			✓					
$f(n) = 7n^5 - 3n^2$							✓	
$f(n) = 99n^3 - 1000n^2$						✓		
$f(n) = n^5 - 99999n^4$							✓	

6)

F – C – E – A – B – D

7)

C = 4

M = 6

$$F(n) = 3n^2 + 5n + 1 = 0 \quad (n^2)$$

C = 1

M = 5

$$F(n) = 3n^2 + 5n + 1 = 0 \quad (n^2)$$

C = 10

M = 2

$$F(n) = 3n^2 + 5n + 1 \neq 0 \quad (n^2)$$

8)

$$C = 1$$

$$M \geq 0$$

$$F(n) = 3n^2 + 5n + 1 = \Omega(n^2)$$

$$C = 1$$

$$M = 5$$

$$F(n) = 3n^2 + 5n + 1 = \Omega(n)$$

$$C = 1$$

$$M = 5$$

$$F(n) = 3n^2 + 5n + 1 \neq \Omega(n)$$

9)

$$C_1 = 1$$

$$C_2 = 4$$

$$M = 6$$

$$F(n) = 3n^2 + 5n + 1 = \Theta(n^2)$$

$$C_1 = 1$$

$$C_2 = 10$$

$$M = 2$$

$$F(n) = 3n^2 + 5n + 1 \neq \Theta(n)$$

$$C_1 = 1$$

$$C_2 = 2$$

$$M \geq 0$$

$$F(n) = 3n^2 + 5n + 1 \neq \Theta(n^3)$$

12)

```
#include <stdio.h>
```

```
void printMaxMin(int array[], int n){
    int max, min;
```

```

if(array[0] > array[1]){
    max = array[0];
    min = array[1];
}
else{
    max = array[1];
    min = array[0];
}

for (int i = 2; i < n; i++){
    if(array[i] > max){
        max = array[i];
    }
    else if(array[i] < min){
        min = array[i];
    }
}
printf("maior: %d\n", max);
printf("menor: %d\n", min);
}

```

13)

a solução mais eficiente é a pesquisa binária pois a ordenação só será necessária uma única vez e após a ordenação o custo de n pesquisas binárias será $n * \Theta(\lg(n))$ e o custo de n pesquisas sequenciais será $n * \Theta(n)$.