

| |
|----------------------|
| Diogo Araujo Miranda |
| Matricula: 705657 |

Exercicio 1)

Tela

Inicio

m1

fim: false

Exercicio 2)

Array A : 1 2 3 4 5 6

Melhor caso, $f(n) = n-1$ comparações (6)

1 > 2 ? false

2 > 3 ? false

3 > 4 ? false

4 > 5 ? false

5 > 6 ? false

Array B : 6 5 4 3 2 1

Pior caso, $(n-1)n/2$ comparações (15)

6 > 5 ? true

5 6 4 3 2 1

tmp = 4

6 > 4 ? true

5 _ 6 3 2 1

5 > 4 ? true

_ 5 6 3 2 1

4 5 6 3 2 1

tmp = 3

6 > 3 ? true

4 5 _ 6 2 1

5 > 3 ? true

4 _ 5 6 2 1

4 > 3 ? true

_ 4 5 6 2 1

3 4 5 6 2 1

temp = 2

6 > 2 ? true

3 4 5 _ 6 1

5 > 2 ? true

3 4 _ 2 6 1

4 > 2 ? true

3 _ 4 2 6 1

3 > 2 ? true

_ 3 4 5 6 1

2 3 4 5 6 1

tmp = 1

6 > 1 ? true

2 3 4 5 _ 6

5 > 1 ? true

2 3 4 _ 5 6

4 > 1 ? true

2 3 _ 4 5 6

3 > 1 ? true

2 _ 3 4 5 6

2 > 1 ? true

_ 2 3 4 5 6

1 2 3 4 5 6

Exercício 2)

12 4 8 2 14 17 6 18 10 16 15 5 13 9 1 11 7 3

temp = 4

12 > 4 ? true

_ 12

4 12

temp = 8

12 > 8 ? true

4 _ 12

4 > 8 ? false

4 8 12

tmp 2

12 > 2 ? true

4 8 _ 12

8 > 2 ? true

4 _ 8 12

4 > 2 ? true

_ 4 8 12
2 4 8 12

tmp 14
12 > 14 ? false
2 4 8 12 14

tmp 17
14 > 17 ? false
2 4 8 12 14 17

tmp 6
17 > 6 ? true
2 4 8 12 14 _ 17
14 > 6 ? true
2 4 8 12 _ 14 17
12 > 6 ? true
2 4 8 _ 12 14 17
8 > 6 ? true
2 4 _ 8 12 14 17
4 > 6 ? false
2 4 6 8 12 14 17

tmp 18
17 > 18 ? false
2 4 6 8 12 14 17 18

tmp 10
18 > 10 ? true
2 4 6 8 12 14 17 _ 18
17 > 10 ? true
2 4 6 8 12 14 _ 17 18
14 > 10 ? true
2 4 6 8 12 _ 14 17 18
12 > 10 ? true
2 4 6 8 _ 12 14 17 18
8 > 10 ? false
2 4 6 8 10 12 14 17 18

tmp 16
18 > 16 ? true
2 4 6 8 10 12 14 17 _ 18
17 > 16 ? true
2 4 6 8 10 12 14 _ 17 18
14 > 16 ? false
2 4 6 8 10 12 14 16 17 18

tmp 15
18 > 15 ? true
2 4 6 8 10 12 14 16 17 _ 18
17 > 15 ? true
2 4 6 8 10 12 14 16 _ 17 18
16 > 15 ? true
2 4 6 8 10 12 14 _ 16 17 18
14 > 15 ? false
2 4 6 8 10 12 14 15 16 17 18

tmp 5
18 > 5 ? true
2 4 6 8 10 12 14 15 16 17 _ 18
17 > 5 ? true
2 4 6 8 10 12 14 15 16 _ 17 18
16 > 5 ? true
2 4 6 8 10 12 14 15 _ 16 17 18

```
15 > 5 ? true
2 4 6 8 10 12 14 _ 15 16 17 18
14 > 5 ? true
2 4 6 8 10 12 _ 14 15 16 17 18
12 > 5 ? true
2 4 6 8 10 _ 12 14 15 16 17 18
10 > 5 ? true
2 4 6 8 _ 10 12 14 15 16 17 18
8 > 5 ? true
2 4 6 _ 8 10 12 14 15 16 17 18
6 > 5 ? true
2 4 _ 6 8 10 12 14 15 16 17 18
4 > 5 ? false
2 4 5 6 8 10 12 14 15 16 17 18
```

```
tmp 13
18 > 13 ? true
2 4 5 6 8 10 12 14 15 16 17 _ 18
17 > 13 ? true
2 4 5 6 8 10 12 14 15 16 _ 17 18
16 > 13 ? true
2 4 5 6 8 10 12 14 15 _ 16 17 18
15 > 13 ? true
2 4 5 6 8 10 12 14 _ 15 16 17 18
14 > 13 ? true
2 4 5 6 8 10 12 _ 14 15 16 17 18
12 > 13 ? false
2 4 5 6 8 10 12 13 14 15 16 17 18
```

```
tmp 9
18 > 9 ? true
2 4 5 6 8 10 12 13 14 15 16 17 _ 18
17 > 9 ? true
2 4 5 6 8 10 12 13 14 15 16 _ 17 18
16 > 9 ? true
2 4 5 6 8 10 12 13 14 15 _ 16 17 18
15 > 9 ? true
2 4 5 6 8 10 12 13 14 _ 15 16 17 18
14 > 9 ? true
2 4 5 6 8 10 12 13 _ 14 15 16 17 18
13 > 9 ? true
2 4 5 6 8 10 12 _ 13 14 15 16 17 18
12 > 9 ? true
2 4 5 6 8 10 _ 12 13 14 15 16 17 18
10 > 9 ? true
2 4 5 6 8 _ 10 12 13 14 15 16 17 18
8 > 9 ? false
2 4 5 6 8 9 10 12 13 14 15 16 17 18
```

```
tmp 1
18 > 1 ? true
2 4 5 6 8 9 10 12 13 14 15 16 17 _ 18
17 > 1 ? true
2 4 5 6 8 9 10 12 13 14 15 16 _ 17 18
16 > 1 ? true
2 4 5 6 8 9 10 12 13 14 15 _ 16 17 18
15 > 1 ? true
2 4 5 6 8 9 10 12 13 14 _ 15 16 17 18
14 > 1 ? true
2 4 5 6 8 9 10 12 13 _ 14 15 16 17 18
13 > 1 ? true
2 4 5 6 8 9 10 12 _ 13 14 15 16 17 18
12 > 1 ? true
2 4 5 6 8 9 10 _ 12 13 14 15 16 17 18
```

```
10 > 1 ? true
2 4 5 6 8 9 _ 10 12 13 14 15 16 17 18
9 > 1 ? true
2 4 5 6 8 _ 9 10 12 13 14 15 16 17 18
8 > 1 ? true
2 4 5 6 _ 8 9 10 12 13 14 15 16 17 18
5 > 1 ? true
2 4 _ 5 6 8 9 10 12 13 14 15 16 17 18
4 > 1 ? true
2 _ 4 5 6 8 9 10 12 13 14 15 16 17 18
2 > 1 ? true
_ 2 4 5 6 8 9 10 12 13 14 15 16 17 18
1 2 4 5 6 8 9 10 12 13 14 15 16 17 18
```

```
tmp 11
18 > 11 ? true
1 2 4 5 6 8 9 10 12 13 14 15 16 17 _ 18
17 > 11 ? true
1 2 4 5 6 8 9 10 12 13 14 15 16 _ 17 18
16 > 11 ? true
1 2 4 5 6 8 9 10 12 13 14 15 _ 16 17 18
15 > 11 ? true
1 2 4 5 6 8 9 10 12 13 14 _ 15 16 17 18
14 > 11 ? true
1 2 4 5 6 8 9 10 12 13 _ 14 15 16 17 18
13 > 11 ? true
1 2 4 5 6 8 9 10 12 _ 13 14 15 16 17 18
12 > 11 ? true
1 2 4 5 6 8 9 10 _ 12 13 14 15 16 17 18
10 > 11 ? false
1 2 4 5 6 8 9 10 11 12 13 14 15 16 17 18
```

```
tmp 7
18 > 7 ? true
1 2 4 5 6 8 9 10 11 12 13 14 15 16 17 _ 18
17 > 7 ? true
1 2 4 5 6 8 9 10 11 12 13 14 15 16 _ 17 18
16 > 7 ? true
1 2 4 5 6 8 9 10 11 12 13 14 15 _ 16 17 18
15 > 7 ? true
1 2 4 5 6 8 9 10 11 12 13 14 _ 15 16 17 18
14 > 7 ? true
1 2 4 5 6 8 9 10 11 12 13 _ 14 15 16 17 18
13 > 7 ? true
1 2 4 5 6 8 9 10 11 12 _ 13 14 15 16 17 18
12 > 7 ? true
1 2 4 5 6 8 9 10 11 _ 12 13 14 15 16 17 18
11 > 7 ? true
1 2 4 5 6 8 9 10 _ 11 12 13 14 15 16 17 18
10 > 7 ? true
1 2 4 5 6 8 9 _ 10 11 12 13 14 15 16 17 18
9 > 7 ? true
1 2 4 5 6 8 _ 9 10 11 12 13 14 15 16 17 18
8 > 7 ? true
1 2 4 5 6 _ 8 9 10 11 12 13 14 15 16 17 18
6 > 7 ? false
1 2 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
```

```
tmp 3
18 > 3 ? true
1 2 4 5 6 7 8 9 10 11 12 13 14 15 16 17 _ 18
17 > 3 ? true
1 2 4 5 6 7 8 9 10 11 12 13 14 15 16 _ 17 18
16 > 3 ? true
```

```

1 2 4 5 6 7 8 9 10 11 12 13 14 15 _ 16 17 18
15 > 3 ? true
1 2 4 5 6 7 8 9 10 11 12 13 14 _ 15 16 17 18
14 > 3 ? true
1 2 4 5 6 7 8 9 10 11 12 13 _ 14 15 16 17 18
13 > 3 ? true
1 2 4 5 6 7 8 9 10 11 12 _ 13 14 15 16 17 18
12 > 3 ? true
1 2 4 5 6 7 8 9 10 11 _ 12 13 14 15 16 17 18
11 > 3 ? true
1 2 4 5 6 7 8 9 10 _ 11 12 13 14 15 16 17 18
10 > 3 ? true
1 2 4 5 6 7 8 9 _ 10 11 12 13 14 15 16 17 18
9 > 3 ? true
1 2 4 5 6 7 8 _ 9 10 11 12 13 14 15 16 17 18
8 > 3 ? true
1 2 4 5 6 7 _ 8 9 10 11 12 13 14 15 16 17 18
7 > 3 ? true
1 2 4 5 6 _ 7 8 9 10 11 12 13 14 15 16 17 18
6 > 3 ? true
1 2 4 5 _ 6 7 8 9 10 11 12 13 14 15 16 17 18
5 > 3 ? true
1 2 4 _ 5 6 7 8 9 10 11 12 13 14 15 16 17 18
4 > 3 ? true
1 2 _ 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
2 > 3 ? false
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

```

Exercicio 3)

```

for(int i = 1; i < SIZE; i++)
{
    int tmp = array[i];
    int j = i - 1;
    int left = 0; // Esquerda da busca binária
    int right = j; // Direita da busca binária
    while( (j >= 0) && (array[j] > tmp))
    {
        int middle = (right+left)/2;
        // Se for maior que meu temporario eu desloco e continuo a busca
pela metade do array
        if(array[middle] >= tmp)
        {
            for(int k = j; k >= middle; k--)
            {
                array[k+1] = array[k];
            }
            j = middle - 1;
            right = j;
        } else {
            // Se não for eu desloco minha busca para o grupo mais a
direita do array
            left = middle + 1;
        }
    }
    array[j+1] = tmp;
}

```

Exercicio 4)

O algoritmo de ordenação por inserção será mais rápido. Isso acontece porque o número de movimentações no algoritmo de ordenação por seleção - independente do caso - sempre será $3(n-1)$, sendo que se o número for igual ao mesmo número, irá executar 3 movimentações que não se justificam. O algoritmo de inserção já tem

uma preocupação maior com o número de movimentações, já que na i -ésima execução ele sempre executa $C_i(n) + 1$ movimentações.