

EXERCÍCIOS INICIAIS

1)

a) $2^0 = 1$

b) $2^1 = 2$

c) $2^2 = 4$

d) $2^3 = 8$

e) $2^4 = 16$

f) $2^5 = 32$

g) $2^6 = 64$

h) $2^7 = 128$

i) $2^8 = 256$

j) $2^9 = 512$

k) $2^{10} = 1024$

l) $2^{11} = 2048$

2)

a) $\lg(2048) = 11$

b) $\lg(1024) = 10$

c) $\lg(512) = 9$

d) $\lg(256) = 8$

e) $\lg(128) = 7$

f) $\lg(64) = 6$

g) $\lg(32) = 5$

h) $\lg(16) = 4$

i) $\lg(8) = 3$

j) $\lg(4) = 2$

k) $\lg(2) = 1$

l) $\lg(1) = 0$

3)

a) $4,01 = 5$

b) $4,01 = 4$

c) $4,99 = 5$

d) $4,99 = 4$

e) $\lg(16) = 4$

f) $\lg(16) = 4$

g) $\lg(17) = 4,087$

h) $\lg(17) = 5$

i) $\lg(17) = 4$

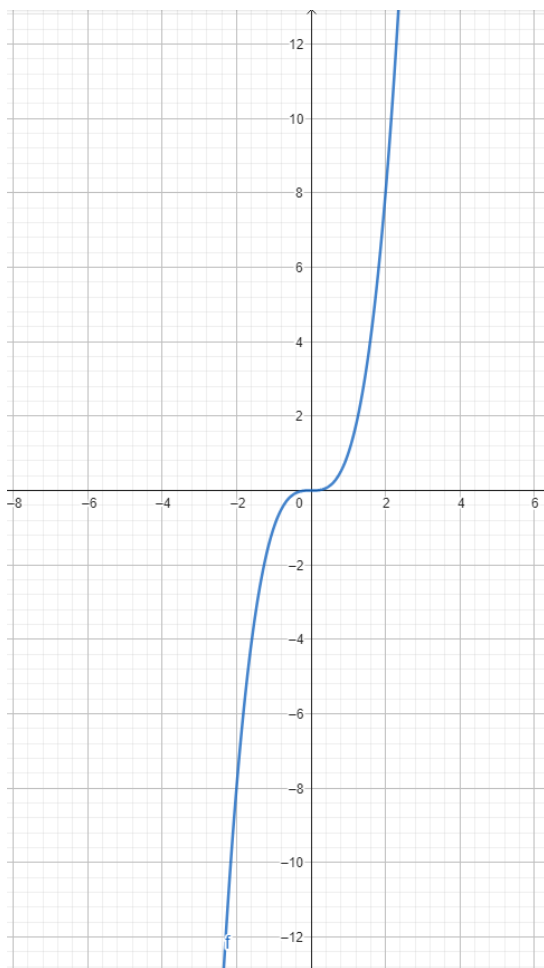
j) $\lg(15) = 3,907$

k) $\lg(15) = 4$

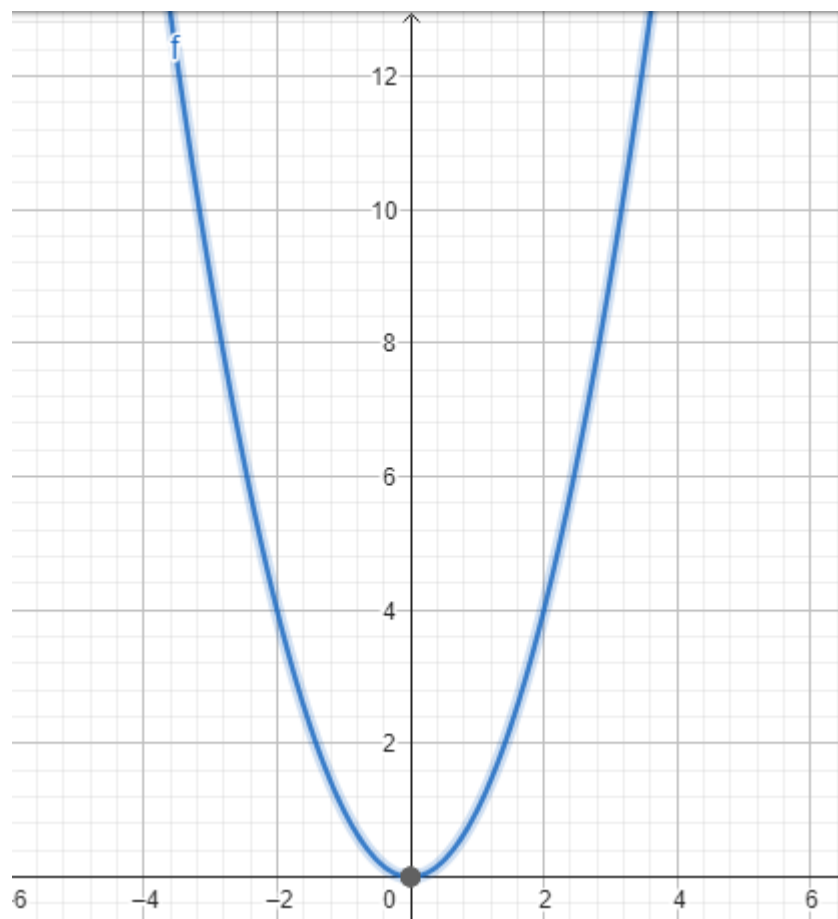
l) $\lg(15) = 3$

4

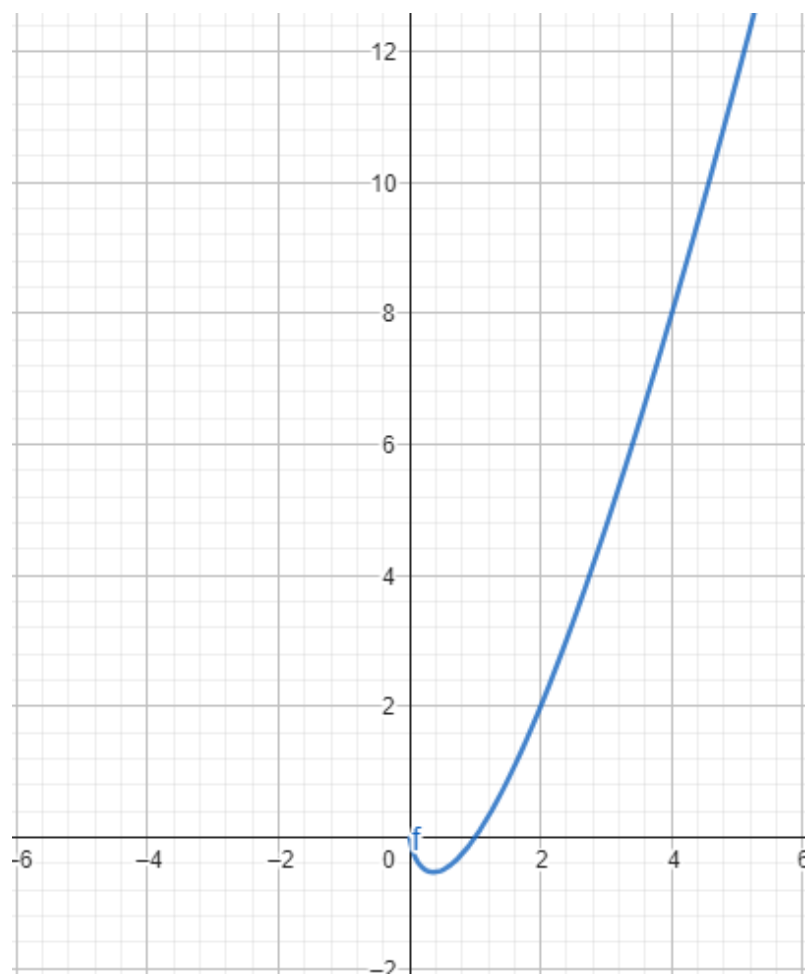
a) $f(n) = n^3$



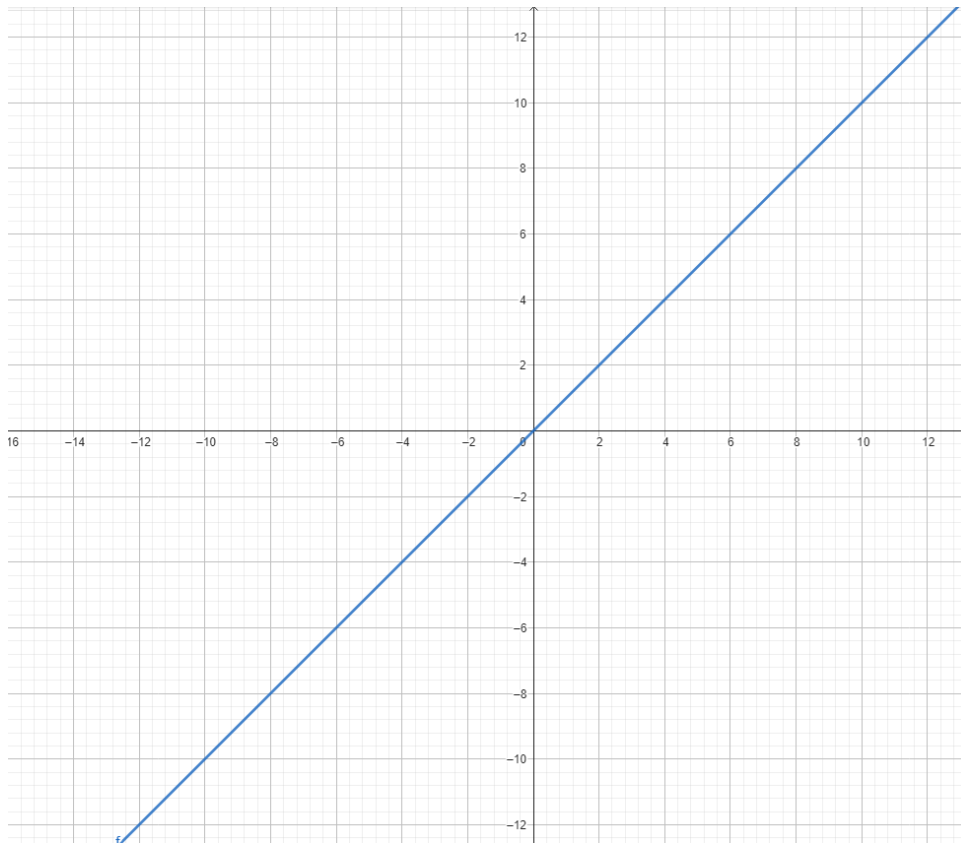
b) $f(n) = n^2$



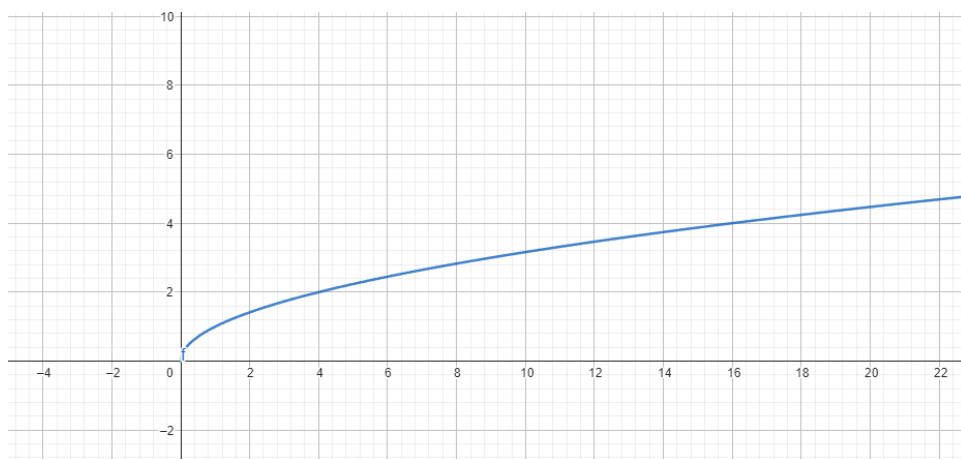
c) $f(n) = n \times \lg(n)$



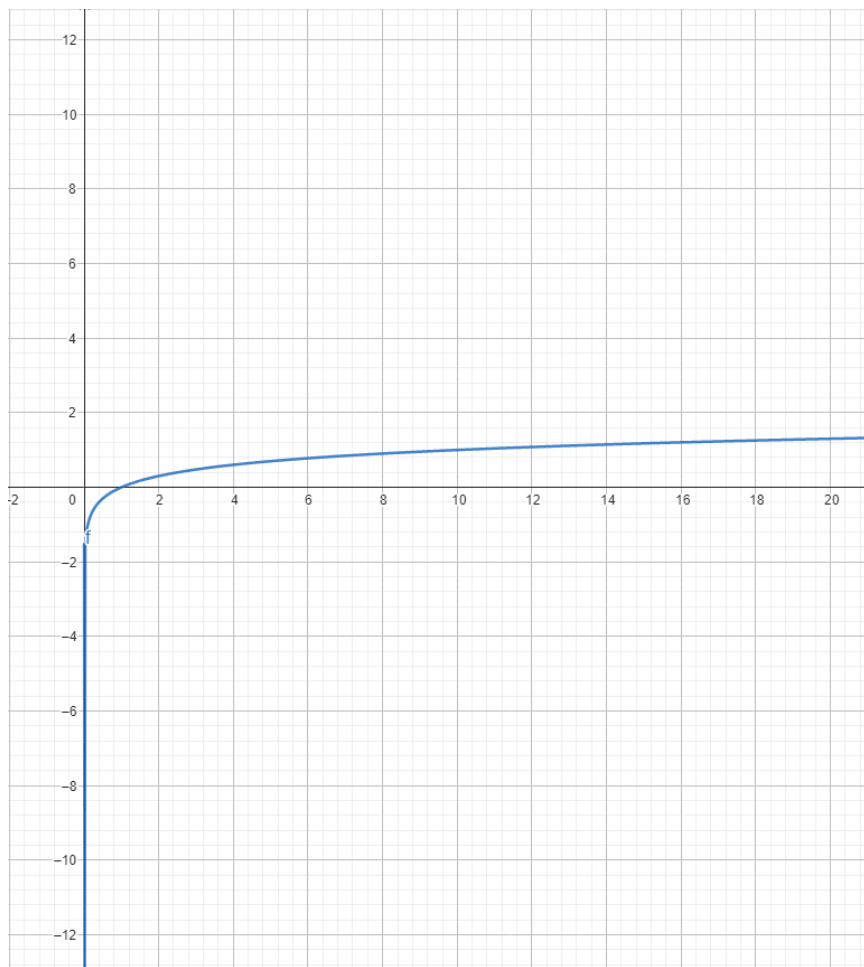
d) $f(n) = n$



e) $f(n) = \text{sqrt}(n)$

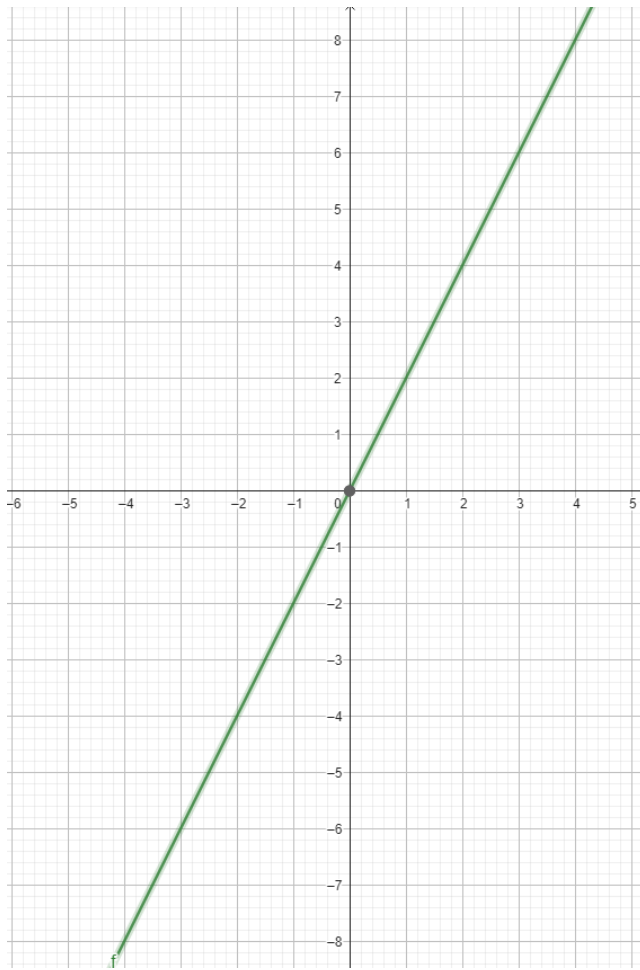


f) $f(n) = \lg(n)$

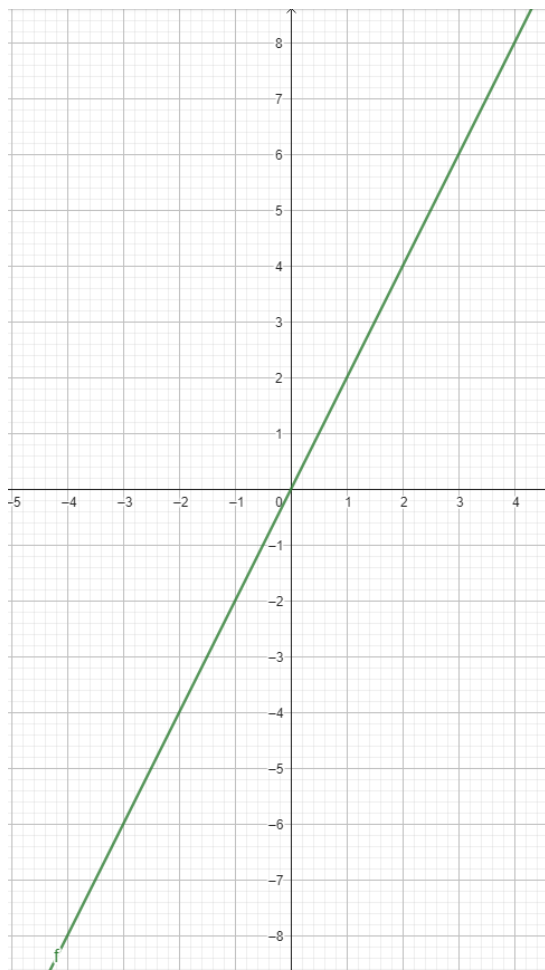


5)

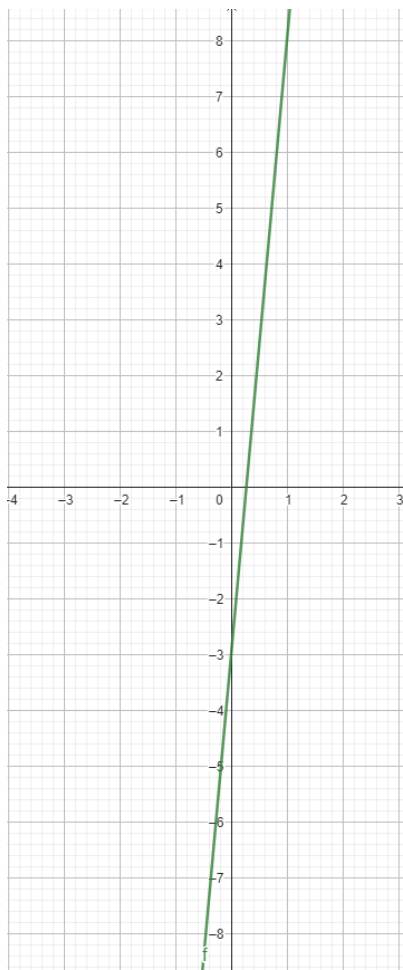
a) $f(n) = n^2$



b) $f(n) = 2 \times | -n^2 |$



c) $f(n) = 3n^2 + 5n - 3$



CONTAGEM DE OPERAÇÕES

6) 3 subtrações

7)

Melhor caso: 3 subtrações

Pior caso: 5 subtrações

8)

Melhor caso (5 subtrações): acontece quando a primeira condição do if é verdadeira ou ambas são falsas

Pior caso (7 subtrações): acontece quando a primeira condição do if é falsa e a segunda, verdadeira.

9) 4 subtrações, quando i vale 0, 1, 2

10) $2n$ subtrações

11) $1 * (n-3)$ subtrações, caso o A fosse 2, por exemplo, o laço faria $2(n-3)$ subtrações

12) 3 subtrações, quando i vale 0, 1, 2

13) 3 subtrações, quando i vale 0, 1, 2

14) 6 subtrações, onde o valor final do laço externo é multiplicado pelo valor final do laço interno, e por fim, multiplicado pela quantidade de subtrações que ocorrem dentro do laço interno

15) $\lg(n) + 1$ subtrações (isso quando n é uma potencia de 2)

16

a) $3n + 2n^2$

```
i = 0;

while (i < n)
{
    i++;
    a--;
    b--;
    c--;
}

for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        a--;
        b--;
    }
}
```

b) $5n + 4n^3$

```
i = 0;
while (i < n){
    i++;
    a--;
    b--;
    c--;
    d--;
    e--;
}
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
```

```

        for (k = 0; k < n; k++, a--, b--, c--, d--);
    }
}

```

c) $\lg(n) + n$

```

i = 1;
while (i < n){
    i *= 2;
    a--;
}
for (i = 0; i < n; i++){
    a--;
}

```

d) $2n^3 + 5$

```

for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        for (k = 0; k < n; k++, a--, b--)
            ;
    }
}
a--;
b--;
c--;
d--;
e--;

```

e) $2n^4 + 2n^2 + n/2$

```
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        a--;
        b--;
        for (k = 0; k < n; k++)
        {
            for (l = 0; l < n; l++, c--; d--)
                ;
        }
    }
    if (i % 2 == 0)
        e--;
}
```

f) $\lg(n) + 5 \lg(n)$

```
i = 1;
while (i < n)
{
    i *= 2;
    a--;
    b--;
    c--;
    d--;
    e--;
    f--;
}
```

FUNÇÕES DE COMPLEXIDADE

17) A operação relevante é a comparação entre elementos do array, e ela será executada se tivermos n elementos.

18)

Melhor caso: Valor encontrado na primeira posição do array

$$t(n) = 1$$

Pior caso: Valor encontrado na ultima posição ou então não existe no array

$$t(n) = n$$

19)

Melhor caso: Valor encontrado na primeira vez que o array é dividido pela metade

$$t(n) = 1$$

Pior caso: elemento desejado não está no array ou está na última posição procurada

$$t(n) = \lg(n)$$

20) O Algoritmo de Seleção realiza $3n-3$ movimentações de registro porque três atribuições são feitas dentro do método swap

21)

```
int mov = 0;
for (int i = 0; i < (n - 1); i++)
{
    int menor = i;
    for (int j = (i + 1); j < n; j++)
    {
        if (array[menor] > array[j])
        {
            menor = j;
        }
    }
    swap(menor, i);
    mov += 3;
}
printf("Teoria: " + (3 * n - 3));
printf("Pratica: " + mov);
```

22) ?

NOÇÕES SOBRE A NOTAÇÃO Θ

23)

Melhor caso: $f(n) = n$, logo, $\Theta(n)$ (quando entra no else)

Pior caso: $f(n) = 2n$, logo, $\Theta(n)$ (quando entra no if)

Exercício

- 1) 4 subtrações, onde $i = 10, 9, 8, 7$
- 2) 4 subtrações, onde $i = 5, 4, 3, 2$
- 3) 8 subtrações, entrará no if quando o i valer 0, 2 e 4, e no else quando o i valer 1 e 3
- 4) 4 subtrações no primeiro loop e mais 8 subtrações no segundo loop, totalizando 12 subtrações
- 5) $n^2 - 3$ multiplicações
- 6) $n^2 - 6$ multiplicações
- 7) $(n-7)^2$ multiplicações
- 8) $\log_2(n) * 1$ multiplicações
- 9) $\log_2(n+1) * 1$ multiplicações
- 10) $\log_2(n) * 1$ multiplicações
- 11) $\log_2(n) * 1 + 1$ multiplicações
- 12) $\log_2(n + 4) * 1$ multiplicações

