

## Noções de Complexidade

1) Resolva as equações abaixo:

- |                    |                      |                      |
|--------------------|----------------------|----------------------|
| 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) Resolva as equações abaixo:

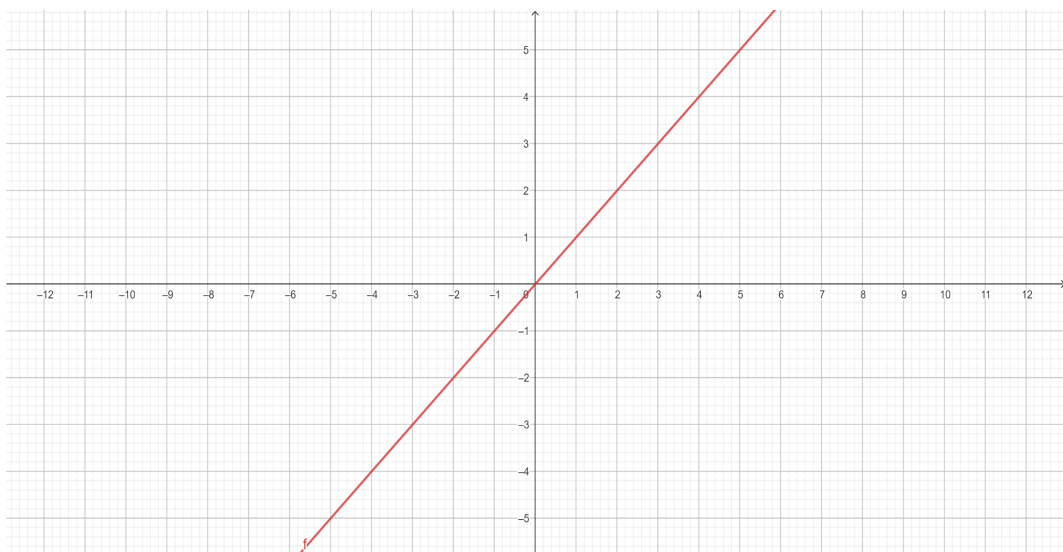
- |                     |                     |                   |
|---------------------|---------------------|-------------------|
| a) $\lg(2048) = 11$ | b) $\lg(1024) = 10$ | c) $\lg(512) = 9$ |
| d) $\lg(256) = 8$   | e) $\lg(128) = 7$   | f) $\lg(64) = 6$  |
| g) $\lg(32) = 5$    | h) $\lg(16) = 4$    | i) $\lg(8) = 3$   |
| j) $\lg(4) = 2$     | k) $\lg(2) = 1$     | l) $\lg(1) = 0$   |

3) Resolva as equações abaixo:

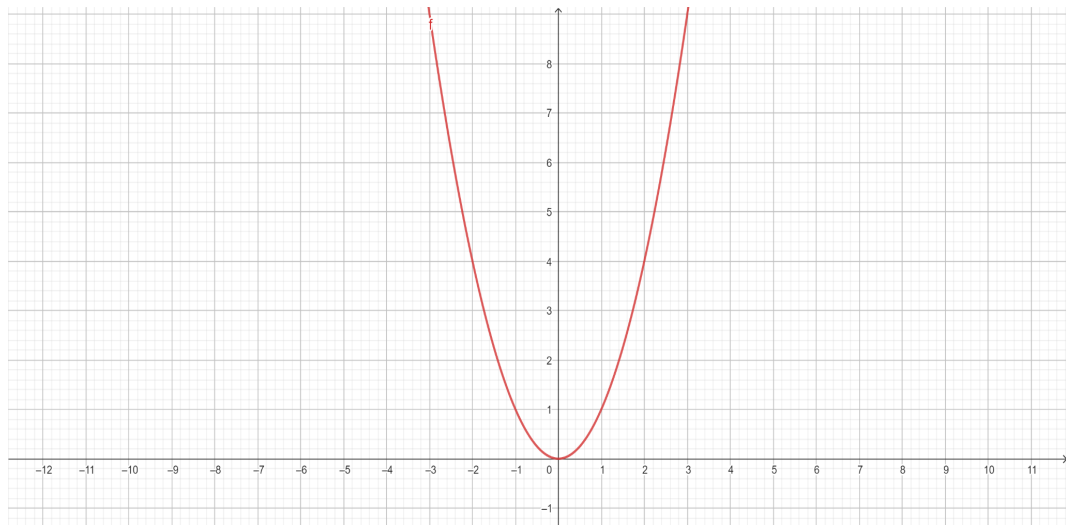
- |                             |                                  |                                  |
|-----------------------------|----------------------------------|----------------------------------|
| a) $\lceil 4,01 \rceil = 5$ | b) $\lfloor 4,01 \rfloor = 4$    | c) $\lceil 4,99 \rceil = 5$      |
| d) $\lceil 4,99 \rceil = 4$ | e) $\lfloor \lg(16) \rfloor = 4$ | f) $\lfloor \lg(16) \rfloor = 4$ |
| g) $\lg(17) = 4.08$         | h) $\lceil \lg(17) \rceil = 5$   | i) $\lfloor \lg(17) \rfloor = 4$ |
| j) $\lg(15) = 3.90$         | k) $\lceil \lg(15) \rceil = 4$   | l) $\lfloor \lg(15) \rfloor = 3$ |

4) Plote um gráfico com todas as funções abaixo:

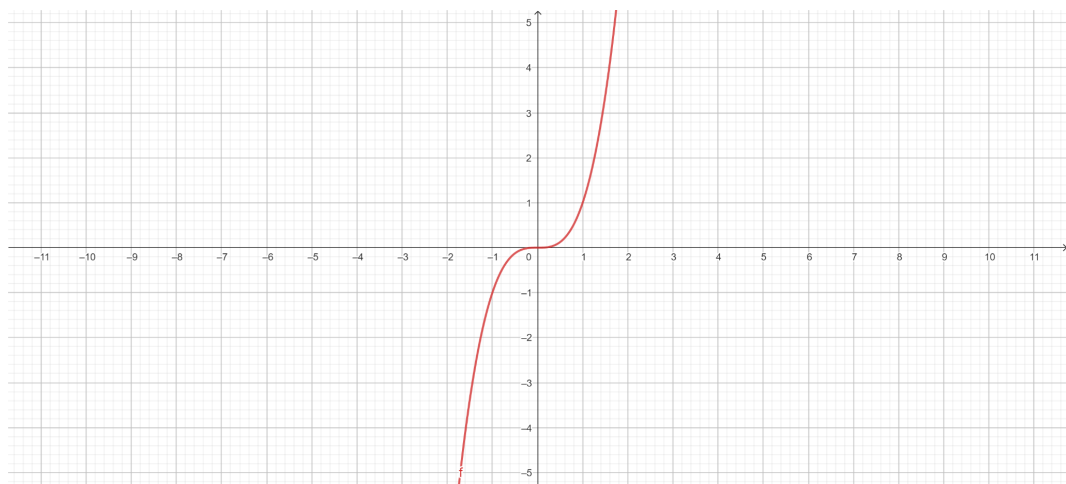
a)  $f(n) = n$



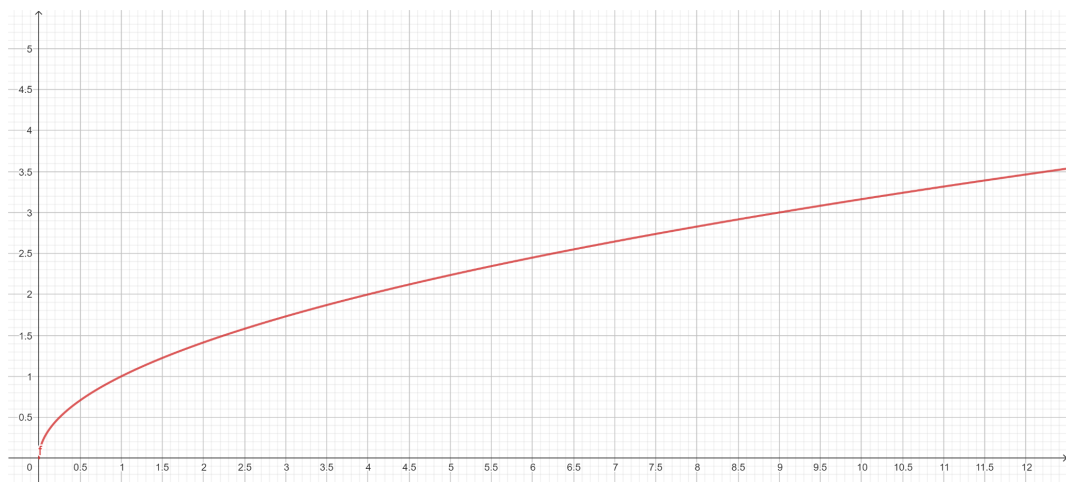
**b)  $f(n) = n^2$**



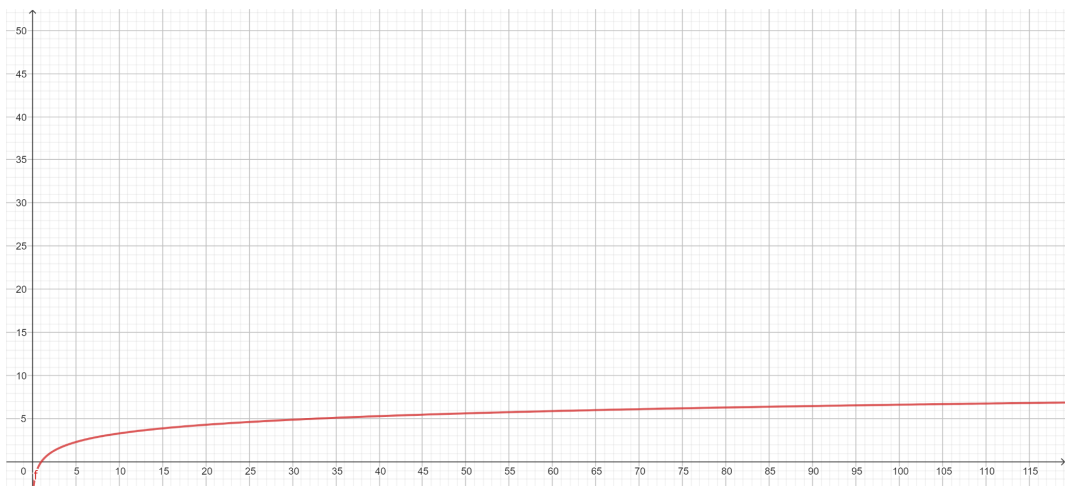
**c)  $f(n) = n^3$**



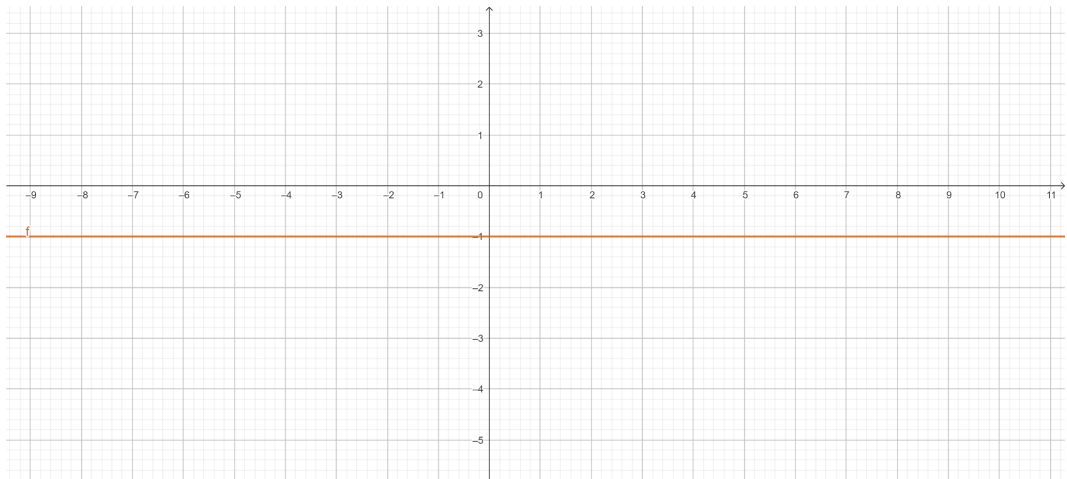
**d)  $f(n) = \sqrt{n}$**



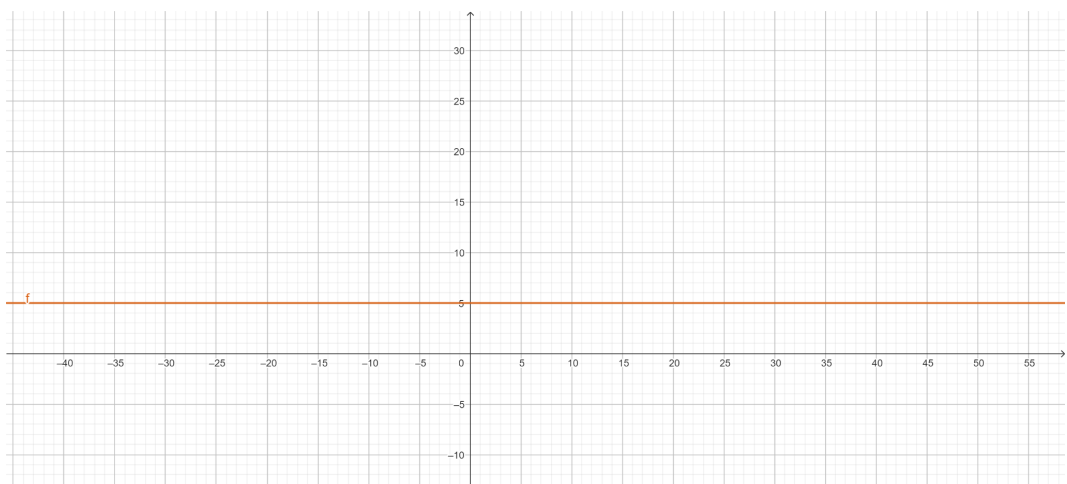
e)  $f(n) = \lg(n) = \log_2(n)$



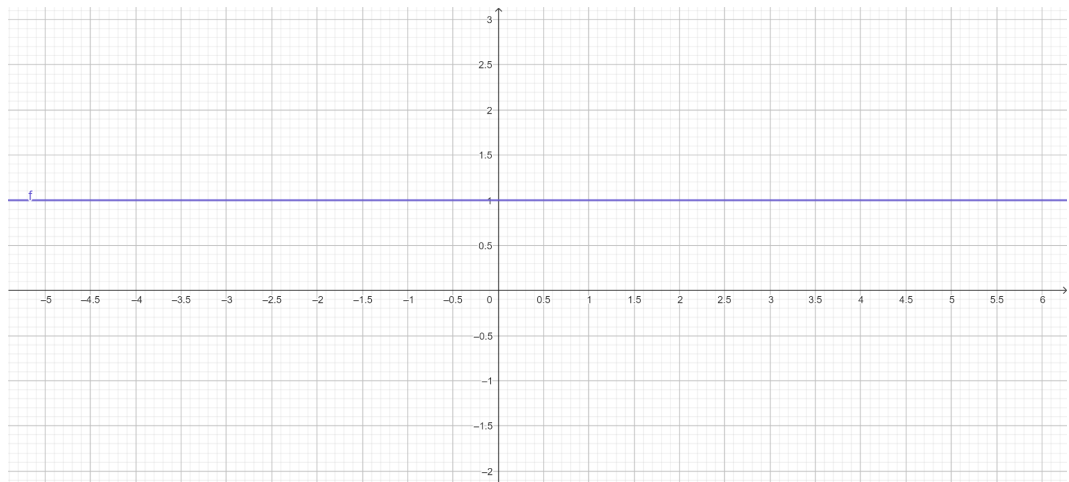
f)  $f(n) = 3n^2 + 5n - 3$



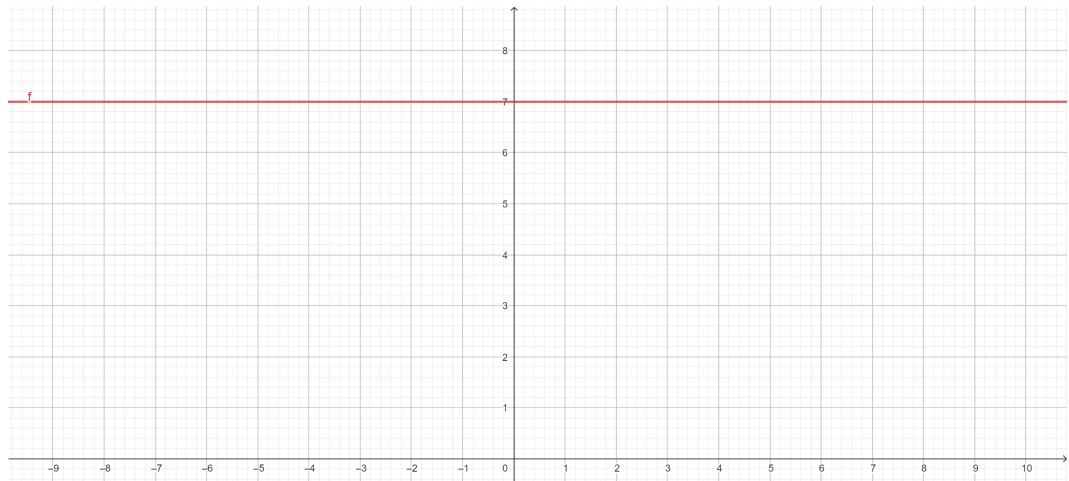
g)  $f(n) = -3n^2 + 5n - 3$



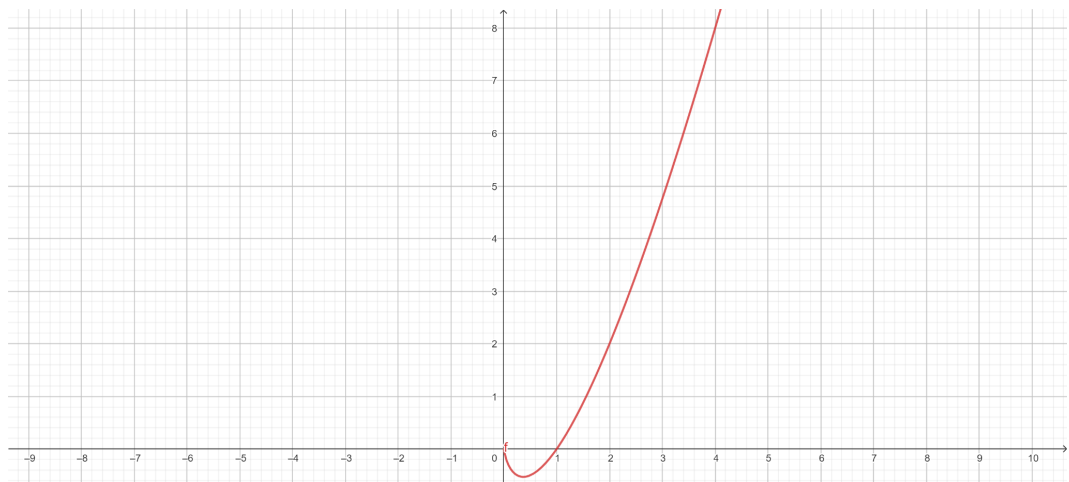
**h)  $f(n) = | - n^2 |$**



**i)  $f(n) = 5n^4 + 2n^2$**



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



## Contagem de Operações

1) Calcule o número de subtrações que o código abaixo realiza:

```
...  
a --;  
a -=;  
a = a -2;  
...
```

São realizadas 3 operações de subtração

2) Calcule o número de adições que o código abaixo realiza:

```
...  
if(a + 5 < b + 3){  
    i ++;  
    ++ b;  
    a += 3;  
}else{  
    j ++;  
}  
...
```

Caso 1:  $(a + 5 < b + 3) == \text{True}$  >> 5 Operações (Pior caso)

Caso 2:  $(a + 5 < b + 3) == \text{False}$  >> 3 Operações (Melhor caso)

3) Calcule o número de adições que o código abaixo realiza:

```
...  
if(a + 5 < b + 3 || c + 1 < d + 3){  
    i ++;  
    ++ b;  
    a += 3;  
}else{  
    j ++;  
}  
...
```

Caso 1:  $(a + 5 < b + 3) == \text{True}$ ,  $(c + 1 < d + 3) == \text{False}$  não realizado  
>> 5 Operações. (Caso Esperado)

Caso 2:  $(a + 5 < b + 3) == \text{False}$ ,  $(c + 1 < d + 3) == \text{True}$   
>> 7 operações. (Pior Caso)

Caso 3:  $(a + 5 < b + 3) == \text{False}$ ,  $(c + 1 < d + 3) == \text{False}$   
>> 5 Operações. (Caso Esperado)

**4) Calcule o número de subtrações que o código abaixo realiza:**

```
...  
for(int i = 0; i < 4; i++){  
    a--;  
}  
...
```

São realizadas 4 operações de subtração.

**5) Calcule o número de subtrações que o código abaixo realiza, resposta em função de n:**

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

Serão realizadas:  $2n$

**6) Calcule o número de subtrações que o código abaixo realiza:**

```
int i = 0, b = 10;  
  
while(i < 3){  
    b--;  
    i++;  
}  
...
```

Serão realizadas um total de 3 operações.

**7) Calcule o número de subtrações que o código abaixo realiza, resposta em função de n:**

```
...  
for(int i = 3; i < n; i++){  
    a--;  
}  
...
```

Caso 1:  $(n \leq 3) \Rightarrow 0$  operações.

Caso 2:  $(n > 3) \Rightarrow n - 3$  operações.

**8) Calcule o número de subtrações que o código abaixo realiza:**

```
int i = 10;
while( i >= 7){
    i--;
}
...
```

São realizadas 4 operações.

**9) Calcule o número de subtrações que o código abaixo realiza:**

```
...
for(int i = 5; i >= 2; i--){
    a--;
}
...
```

São realizadas 8 operações.

**10) Calcule o número de subtrações que o código abaixo realiza:**

```
...
for(int i = 0; i < 5; i++){
    if(i % 2 == 0){
        a--;
        b--;
    }else{
        c--;
    }
}
...
```

São realizadas 8 operações.

**11) Calcule o número de subtrações que o código abaixo realiza:**

```
int a = 10;
for(int i = 0; i < 3; i++){
    for(int j = 0; j < 2; j++){
        a--;
    }
}
...
```

São realizadas 6 operações;

**12) Calcule o número de subtrações que o código abaixo realiza:**

**int a = 10, b = 10, c = 10, d = 10;**

```
for(int i = 0; i < 3; i++){  
    for(int j = 0; j < 2; j++){  
        a--;  
        b--;  
        c--;  
        d--;  
    }  
}  
...
```

São realizadas 24 operações;

**13) Calcule o número de subtrações que o código abaixo realiza:**

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

São realizadas  $n^2$  operações;

**14) Calcule o número de subtrações que o código abaixo realiza:**

```
int i = 1, b = 10;  
while(i > 0){  
    b--;  
    i = i >> 1;  
}  
i = 0  
while( i < 15){  
    b--;  
    i += 2;  
}  
...
```

Serão realizadas 8 operações.



**15) Calcule o número de multiplicações que o código abaixo realiza:**

```
...
for(int i = 0; i < n; i++)
    for(int j = 0; j < n - 3; j++)
        a *= 2;
...
```

Serão realizadas  $(n - 3) * n$  ou  $n^2 - 3n$  operações;

**16) Calcule o número de multiplicações que o código abaixo realiza:**

```
...
for(int i = n - 7; i > 1; i--)
    for(int j = 0; j < n; j++)
        a *= 2;
...
```

Serão realizadas  $(n - 8) * n$  ou  $n^2 - 8n$  operações;

**17) Calcule o número de multiplicações que o código abaixo realiza:**

```
...
for (int i = n; i > 0; i /= 2)
    a *= 2;
...
```

Serão realizadas  $\lfloor \lg(n) \rfloor + 1$  operações;

**18) Calcule o número de multiplicações que o código abaixo realiza:**

```
...
for (int i = n+4; i > 0; i >>= 1)
    a *= 2;
...
```

Serão realizadas  $\lfloor \lg(n) \rfloor + 1$  operações;

**19) Calcule o número de multiplicações que o código abaixo realiza:**

```
...
for (int i = n - 7; i >= 1; i--)
    for (int j = n - 7; j >= 1; j--)
        a *= 2;
...
```

Serão realizadas  $(n - 8)(n - 8)$  operações;

**20) Calcule o número de multiplicações que o código abaixo realiza:**

```
...
for (int i = n + 1; i > 0; i /= 2)
    a *= 2;
...
```

Serão realizadas  $\lfloor \lg(n) \rfloor + 1$  operações;

**21) Calcule o número de multiplicações que o código abaixo realiza:**

```
...
for (int i = n; i > 1; i /= 2)
    a *= 2;
...
```

Serão realizadas  $\lfloor \lg(n) \rfloor$  operações;

**22) Calcule o número de multiplicações que o código abaixo realiza:**

```
...
for (int i = 1; i < n; i *= 2)
    a *= 2;
...
```

Serão realizadas  $2(\lfloor \lg(n) \rfloor + 1) + 1$  relevantes ou  $\lfloor \lg(n) \rfloor + 1$  operações;

**23) Calcule o número de multiplicações que o código abaixo realiza: <<<<<< Revisar**

```
...
for (int i = 1; i <= n; i *= 2)
    a *= 2;
...
```

Serão realizadas  $2(\lfloor \lg(n) \rfloor + 1) + 1$  relevantes ou  $\lfloor \lg(n) \rfloor + 1$  operações;

**24) Faça um método que receba um número inteiro n e efetue o número de subtrações pedido em:**

**a)  $3n + 2n^2$**

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

**b)  $5n + 4n^3$**

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

**c)  $\lg(n) + n$**

```
...
for(int i = n; i > 1; i /= 2){
    a--;
}

for(int j = 0; j < n; j++){
    a--;
}
...
```

**d)  $2n^3 + 5$**

```
...
int x = 5;

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

while(x > 0){
    x--;
}
...
```

**e)  $9(n^2 * n^2) + 5n^2 + n/2$**

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

    int b = 5;

    while(b > 0){
        b--;
    }

    for(int j = 0; j < (n * n); j++){
        for(int k = 0; k < 9; k++) a--;
    }
}

int x = n/2;

while(x > 0) x--;
...
```

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

```
...
int i = 5;
for(int i = 1; i < n; i *= 2) a--;
while(i > 0){
    for(int j = n; j > 0; j /= 2) a--;
}
...
```

**25) Encontre o menor valor em um array de inteiros.**

```
int min = array[0];

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

**a) Encontre a operação relevante:**  $\text{min} > \text{array}[i]$

**b) Número de vezes em que é executada:**  $f(n) = n - 1$