

Nome: Victor Ferraz de Moraes
Curso: Ciência da Computação
Matrícula: 802371

EXERCÍCIOS RESOLVIDOS:

Exercício 1:

Exercício Resolvido (1): Resolva as Equações

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

PUC Minas Virtual

Exercício 2:

Exercício Resolvido (2): Resolva as Equações

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

PUC Minas Virtual

Exercício 3:

Exercício Resolvido (3): Resolva as Equações

a) $\lceil \overline{4,01} \rceil = 5$ d) $\lfloor \overline{4,99} \rfloor = 4$ g) $\lg(17) = 4,087$ j) $\lg(15) = 3,90$

b) $\lfloor \overline{4,01} \rfloor = 4$ e) $\lceil \lg(16) \rceil = 4$ h) $\lceil \lg(17) \rceil = 5$ k) $\lceil \lg(15) \rceil = 4$

c) $\lfloor \overline{4,99} \rfloor = 5$ f) $\lfloor \lg(16) \rfloor = 4$ i) $\lfloor \lg(17) \rfloor = 4$ l) $\lfloor \lg(15) \rfloor = 3$

PUC Minas Virtual

Exercício 4:

Exercício Resolvido (4): Plote os Gráficos

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

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

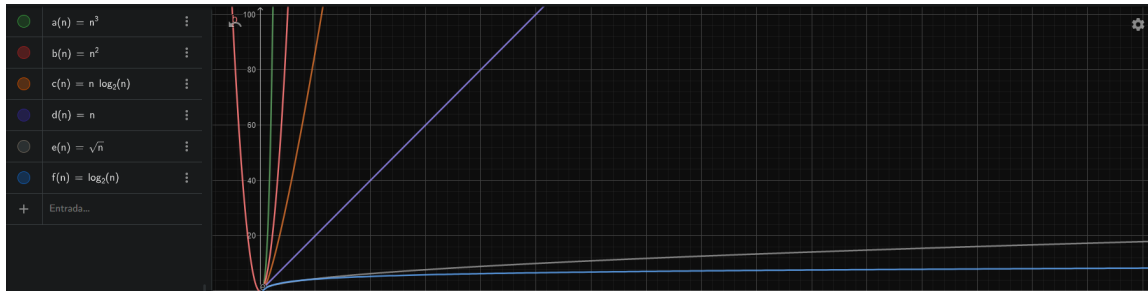
c) $f(n) = n \times \lg(n)$

d) $f(n) = n$

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

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

PUC Minas Virtual



Exercício 5:

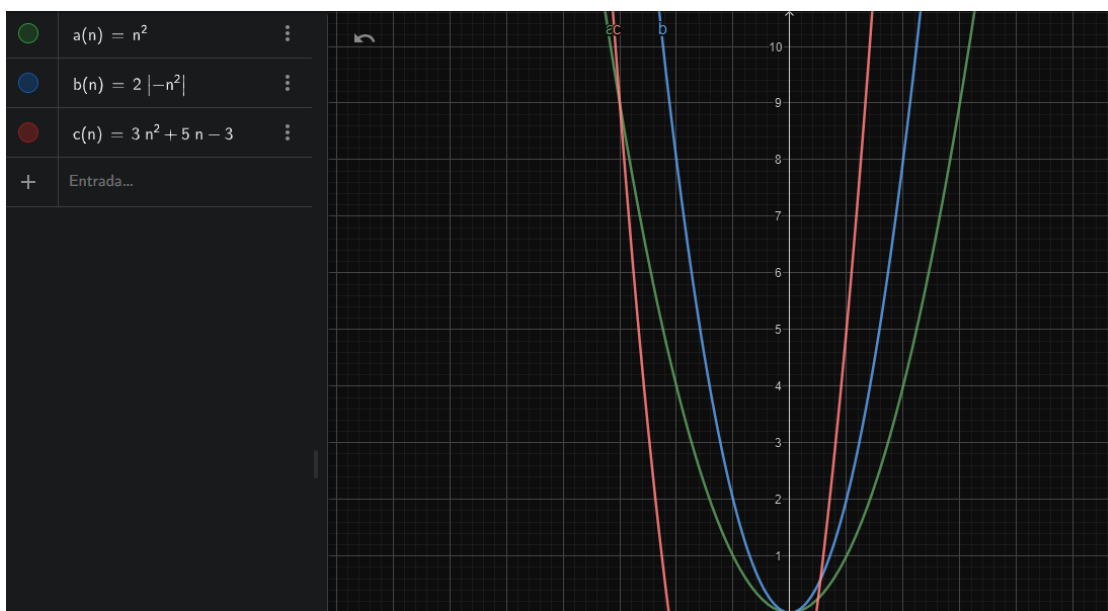
Exercício Resolvido (5): Plote os Gráficos

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

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

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

PUC Minas Virtual



Exercício 6:

Exercício Resolvido (6)

- Calcule o **número de subtrações** que o código abaixo realiza:

```
...  
a--; 1  
a -= 3; 2  
a = a - 2; 3
```

3 subtrações

PUC Minas Virtual

Exercício 7:

Exercício Resolvido (7)

- Calcule o **número de subtrações** que o código abaixo realiza:

```
... 1 1 2 2  
if (a - 5 < b - 3){  
    i--; 3  
    --b; 4  
    a -= 3; 5  
} else {  
    j--; 3  
}
```

se verdadeiro:
5 subtrações

se falso:
3 subtrações

PUC Minas Virtual

Exercício 8:

Exercício Resolvido (8)

- Calcule o **número de subtrações** que o código abaixo realiza:

```
... 11 22 33 44  
if (a - 5 < b - 3 || c - 1 < d - 3){  
    i--; 5  
    --b; 6  
    a -= 3; 7  
} else {  
    j--; 5  
}
```

se verdadeiro:
4 subtrações
se falso:
5 subtrações

PUC Minas Virtual

Exercício 9:

Exercício Resolvido (9)

- Calcule o **número de subtrações** que o código abaixo realiza:

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

} código repete 4x

4 subtrações

PUC Minas Virtual

Exercício 10:

Exercício Resolvido (10)

- Calcule o **número de subtrações** que o código abaixo realiza:

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

Observação: Sua resposta deve ser em função de n

O código realiza $2n$ subtrações

PUC Minas Virtual

Exercício 11:

Exercício Resolvido (11)

- Calcule o **número de subtrações** que o código abaixo realiza:

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

O código realiza $n - 3$ subtrações

PUC Minas Virtual

Exercício 12:

Exercício Resolvido (12)

- Calcule o **número de subtrações** que o código abaixo realiza:

```
...  
int i = 0, b = 10;
```

```
while (i < 3){  
    i++;  
    b--;  
}
```

repete 3x

1 subtração

$1 \cdot 3 = 3$ subtrações

PUC Minas Virtual

Exercício 13:

Exercício Resolvido (13)

- Calcule o **número de subtrações** que o código abaixo realiza: ✓

```
...  
int i = 0, b = 10;
```

```
do {  
    i++;  
    b--;  
} while (i < 3);
```

← subtração inicial

3 subtrações

PUC Minas Virtual

Exercício 14:

Exercício Resolvido (14)

- Calcule o **número de subtrações** que o código abaixo realiza:

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

} O número de subtrações será
 $3 \cdot 2 = 6$ subtrações

PUC Minas Virtual

Exercício 15:

Exercício Resolvido (15)

- Calcule o **número de multiplicações** que o código abaixo realiza:

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

PUC Minas Virtual

Supondo que N tenha o valor de 8, ele irá realizar 4 multiplicações. Se N for 4, irá realizar 3 multiplicações. Portanto, o código realiza $\lg(n) + 1$ multiplicações.

Exercício 16:

Exercício Resolvido (16)

- Faça um método que receba um número inteiro n e efetue o **número de subtrações** pedido em:
 - a) $3n + 2n^2$
 - b) $5n + 4n^3$
 - c) $\lg(n) + n$
 - d) $2n^3 + 5$
 - e) $2n^4 + 2n^2 + n/2$
 - f) $\lg(n) + 5 \lg(n)$

PUC Minas Virtual

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

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

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

```

        e--;
    }

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

```

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

d) for (int i = 0; i < n; i++){
 for (int j = 0; j < n; j++){
 for (int k = 0; k < n; k++){
 a--;
 b--;
 }
 }
 }
 for (int i = 0; i < 5; i++){
 a--;
 }

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

```

        for (int j = 0; j < n; j++){
            a--;
            b--;
        }
    }
    for (int i = 0; i < n/2; i++){
        a--;
    }

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

```

Exercício 17:

Exercício Resolvido (17)

- 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];
    }
}

```

PUC Minas Virtual

- A operação mais relevante seria comparar os elementos do Array
- Ela será executada todas as vezes que o if for executado, ou seja, n-1 vezes
- Todos os casos ela será executada.

d) Sim, pois ele executa o mínimo de ações possível.

Exercício 18:

Exercício Resolvido (18)

- Apresente a função de complexidade de tempo (número de comparações entre elementos do *array*) da pesquisa sequencial no melhor e no pior caso

```
boolean resp = false;

for (int i = 0; i < n; i++){
    if (array[i] == x){
        resp = true;
        i = n;
    }
}
```

PUC Minas Virtual

Melhor Caso:

$t(n) = 1$, pois o número estará na primeira posição do array.

Pior Caso:

$t(n) = n$, pois o número estará em alguma posição além da primeira do array.

Exercício 19:

Exercício Resolvido (19)

- Apresente a função de complexidade de tempo (número de comparações entre elementos do *array*) da pesquisa binária no melhor e no pior caso

```
boolean resp = false;
int dir = n - 1, esq = 0, meio, diferenca;
while (esq <= dir) {
    meio = (esq + dir) / 2;
    diferenca = (x - array[meio]);
    if (diferenca == 0){
        resp = true;
        esq = n;
    } else if (diferenca > 0){
        esq = meio + 1;
    } else {
        dir = meio - 1;
    }
}
```

PUC Minas Virtual

Melhor Caso:

$t(n) = 1$, pois a condição “diferença = $(x - \text{array}[\text{meio}]);$ ” vai ser verdadeira.

Pior Caso:

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

Exercício 20:

Exercício Resolvido (20)

- Explique porque o Algoritmo de Seleção realiza $m(n) = 3n - 3$ movimentações de registros

```
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);  
}
```

```
void swap(int a, int b) {  
    int temp = array[a];  
    array[a] = array[b];  
    array[b] = temp;  
}
```

PUC Minas Virtual

Pois, o algoritmo itera $n-1$ vezes e para cada iteração, o swap realiza 3 movimentações, ou seja, $3 \text{ movimentações} * (n-1) = 3n - 3$.

Exercício 21:

Exercício Resolvido (21)

- Modifique o código do Algoritmo de Seleção para que ele contabilize o número de movimentações de registros

```
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);  
}
```

```
int movimentacoes = 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);  
    movimentacoes+=3;  
}
```

Exercício 22:

Exercício Resolvido (22)

- Explique porque o Algoritmo de Seleção realiza $c(n) = \frac{n^2}{2} - \frac{n}{2}$ comparações entre registros

```
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);  
}
```

PUC Minas Virtual

Para explicar a fórmula fornecida, é necessário saber os somatórios. Como não possuo esse conhecimento, não consegui resolver a questão.

Exercício 23:

Exercício Resolvido (23)

- Calcule o **número de subtrações** que o código abaixo realiza:

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

PUC Minas Virtual

O número de subtrações em ambos os casos, melhor ou pior irá ser $\Theta(n)$, pois em ambos os casos, o número de subtrações é uma constante. Sendo o melhor caso $f(n) = n$ e o pior caso $f(n) = 2n$.

EXERCÍCIOS:

Exercício 1:

Exercício (1)

- Calcule o **número de subtrações** que o código abaixo realiza:

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

O código realiza 4 subtrações, logo, $\Theta(1)$.

Exercício 2:

Exercício (2)

- Calcule o **número de subtrações** que o código abaixo realiza:

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

O código realiza 4 subtrações na lista. Se contarmos o decremento de i , serão 9 subtrações no total, logo, $\Theta(1)$.

Exercício 3:

Exercício (3)

- 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--;
    }
}
```

PUC Minas Virtual

O código irá realizar 7 subtrações, pois em momentos que a condição “ $i \% 2 == 0$ ” for verdadeira, o código irá realizar duas subtrações. Essa condição é verdadeira quando i assume o valor de 2 e 4, logo o número total de subtrações é respectivamente: $1 + 1 + 2 + 1 + 2 = 7$.

Exercício 4:

Exercício (4)

- Calcule o **número de subtrações** que o código abaixo realiza:

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

PUC Minas Virtual

O número de subtrações nesse código é $4 + 7 = 11$ subtrações, pois no primeiro while o custo da iteração é $\lg(i) + 1$ e no segundo while i está incrementando de 2 e a condição se torna falsa somente no oitavo incremento.

Exercício 5:

Exercício (5)

- 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;
    }
}
```

PUC Minas Virtual

O código realiza $n(n-3)$ multiplicações. Pois para cada laço interno, ele realiza uma multiplicação. Ou seja, n repetições (laço externo) * $(n-3)$ repetições (laço interno) * 1 multiplicação = $n(n-3)$ multiplicações.

Exercício 6:

Exercício (6)

- 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;
    }
}
```

PUC Minas Virtual

O código realiza $n(n-7)$ multiplicações, pois o laço externo realiza $n-7$ repetições e o laço interno realiza n repetições, gerando uma multiplicação para cada laço interno, ou seja, $n(n-7)$.

Exercício 7:

Exercício (7)

- 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;  
    }  
}
```

PUC Minas Virtual

O código realiza $(n-7)^2$ repetições, pois o laço externo realiza $n-7$ repetições e o laço interno realiza $n-7$ repetições, gerando uma multiplicação para cada laço interno, ou seja, $(n-7)^2$.

Exercício 8:

Exercício (8)

- Calcule o **número de multiplicações** que o código abaixo realiza:

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

PUC Minas Virtual

O número de multiplicações vai ser $\lg(n)$, pois para cada repetição o valor de i se divide por 2 e realiza 1 multiplicação. Então por exemplo, se i valesse 8, ele iria valer 4 na segunda iteração, depois 2 na terceira iteração e então as repetições terminariam quando ele se dividisse por 2 pela quarta vez resultando em 1, fazendo com que esse algoritmo tenha um custo logarítmico.

Exercício 9:

Exercício (9)

- Calcule o **número de multiplicações** que o código abaixo realiza:

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

PUC Minas Virtual

O algoritmo terá $\lg(n+1) + 1$ multiplicações.

Exercício 10:

Exercício (10)

- Calcule o **número de multiplicações** que o código abaixo realiza:

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

PUC Minas Virtual

O algoritmo terá $\lg(n)$ multiplicações.

Exercício 11:

Exercício (11)

- Calcule o **número de multiplicações** que o código abaixo realiza:

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

PUC Minas Virtual

O algoritmo terá $\lg(n) + 1$ multiplicações.

Exercício 12:

Exercício (12)

- Calcule o **número de multiplicações** que o código abaixo realiza:

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

PUC Minas Virtual

O algoritmo terá $\lg(n+4) + 1$ multiplicações.