RESOLVA AS EQUAÇÕES

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) $\lceil 4{,}01 \rceil = 5$ b) $\lfloor 4{,}01 \rfloor = 4$ c) $\lceil 4{,}99 \rceil = 5$

d) $\lfloor 4{,}99 \rfloor = 4$ e) $\lceil lg(16) \rceil = 4$ 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)

$f(n) = n^3$ ~> - - -

$f(n) = n^2$ ~> . . . .

$f(n) = n \cdot lg(n)$ ~> ▷▷▷

$f(n) = n$ ~> ~~AA~~

$f(n) = sqrt(n)$ ~> +++++

$f(n) = lg(n)$ ~> ——

CONTAGEM DE OPERAÇÕES

1) Calcule o número de subtrações do código abaixo:

...

```
a--;
a-=3;
a = a-2;
```

→ 3

b)

```
if(a-5 > b-3){
    a--;
    --b;
    a-=3;
}else{
    j--;
}
```

→ MELHOR CASO 3

PIOR CASO 5

c)

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

→ MELHOR CASO 3

PIOR CASO 7

d)

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

→ 4

E) for(int i = 0; i < N; i++){
a--;
b--;
}
$\rightarrow$ N x 2

F) for(int i = 3; i < N; i++){
a--;
}
$\rightarrow$ N - 3

G) int i = 0, b = 10

while(i < 3){
i++;
b--;
}
$\rightarrow$ 3 VEZES

H) int i = 0, b = 10;

do {
i++;
b--;
}while (i < 3);
$\rightarrow$ 3 VEZES

I) for (int i = 0; i < 3; i++){
for(int j = 0; j < 2; j++){
a--;
}
}
$\rightarrow$ 6

2) CALCULE O NÚMERO DE MULTIPLICAÇÕES QUE O CÓDIGO ABAIXO REALIZA:

a)
```
for(int i = n; i > 0; i /= 2){
    a *= 2;
}
```
"log(n) + 1 VEZES"

3) FAÇA UM MÉTODO QUE RECEBA UM NÚMERO INTEIRO n E EFETUE O NÚMERO DE SUBTRAÇÕES PEDIDO EM:

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)

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

C) D) E) F) Não consegui realizar os exercícios.

- 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];
    }
}
```

1º) Qual é a operação relevante?

2º) Quantas vezes ela será executada?

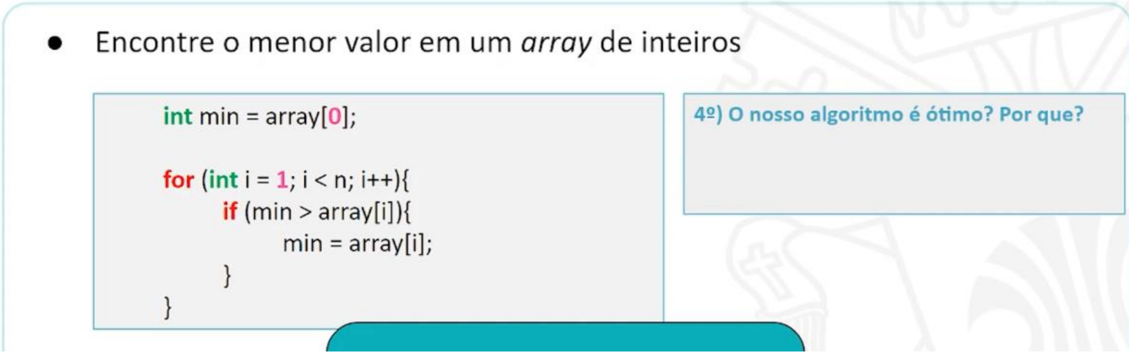
1) A operação relevante é a comparação de min > array[i]
2) N-1

- Da mesma forma que calculamos o custo de um churrasco:
  - Carne: 400 gramas por pessoa (preço médio do kg R$ 20,00 - picanha, asinha, coraçãozinho ...)
  - Cerveja: 1,2 litros por pessoa (litro R$ 3,80)
  - Refrigerante: 1 litro por pessoa (Garrafa 2 litros R$ 3,50)

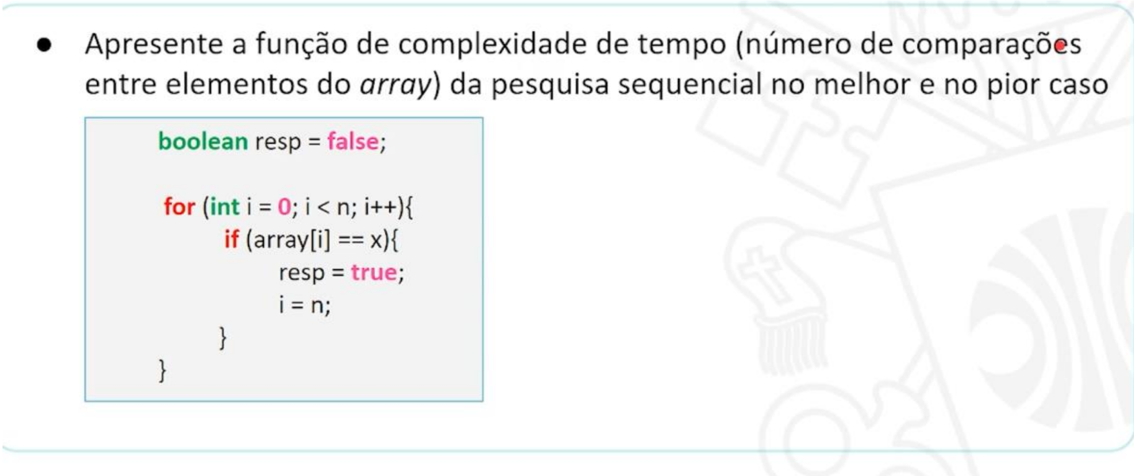
Exercício: Monte a função de complexidade (ou custo) do nosso churrasco

RESPOSTA:

N * 400/1000 * 20 + N * 1200/1000 * 3,8 + N * 1 * 3,50/2



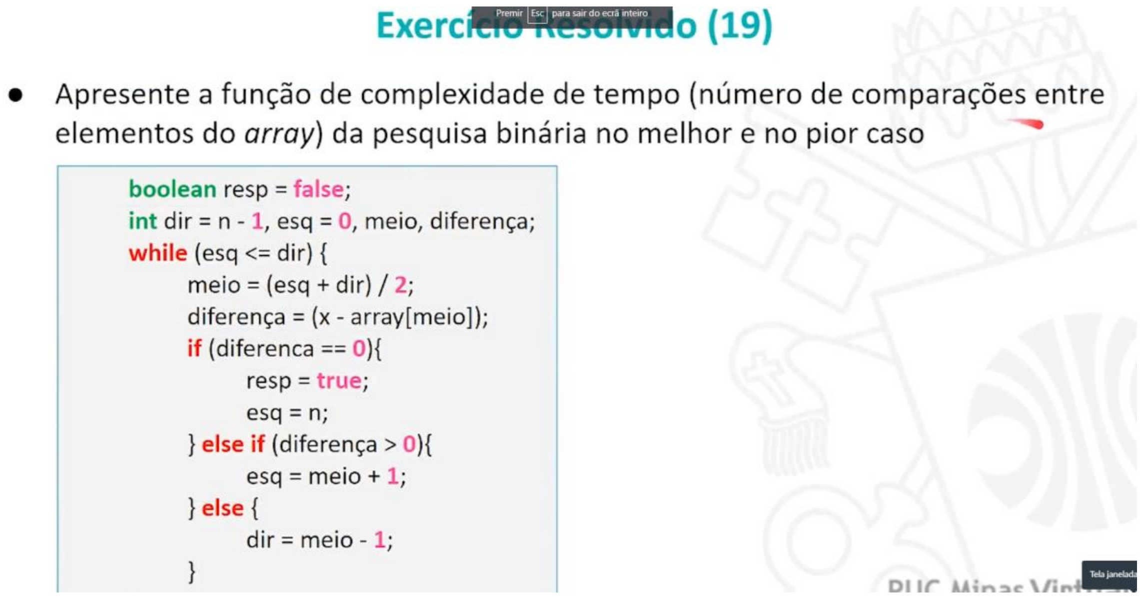
RESPOSTA: Sim, porque precisamos comparar com todos os elementos do array para garantir que encontremos o menor.



RESPOSTA:

MELHOR CASO: 1

PIOR CASO: N



MELHOR CASO: Elemento buscado se encontra no meio do vetor.

PIOR CASO: Elemento não está no vetor, ou está na última posição procurada.

- 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;
}
```

RESPOSTA: Porque a cada repetição ele executa 3 movimentos, temos como quantidade de repetições n-1, essa -1 repetição equivale a -3 movimentos, resultando em 3n-3.

- 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);
}
```

<u>RESPOSTA:</u>

```
int cont = 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;
        }
    }
    cont += 3;
    swap(menor, j);
}
```

- 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);
}
```

RESPOSTA: Não compreendi

$$c(n) = \sum_{i=0}^{n-2} (n - i - 1)$$

-

NOÇÕES SOBRE A NOTAÇÃO Θ

- 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--;
        }
}
```

RESPOSTA:

Melhor Caso: f(n) = n

Pior Caso: f(n) = 2n

Ou seja, $\Theta(n)$.