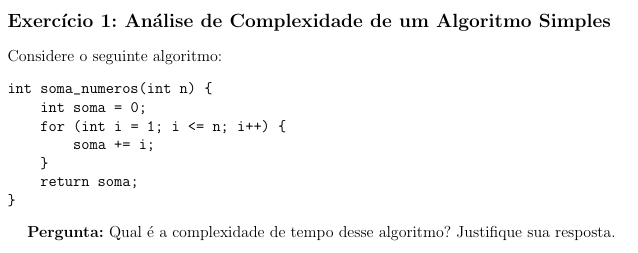
# RESOLUÇÃO DA LISTA 3



Resposta Questão 1:

C1 + C2 \* n + C3

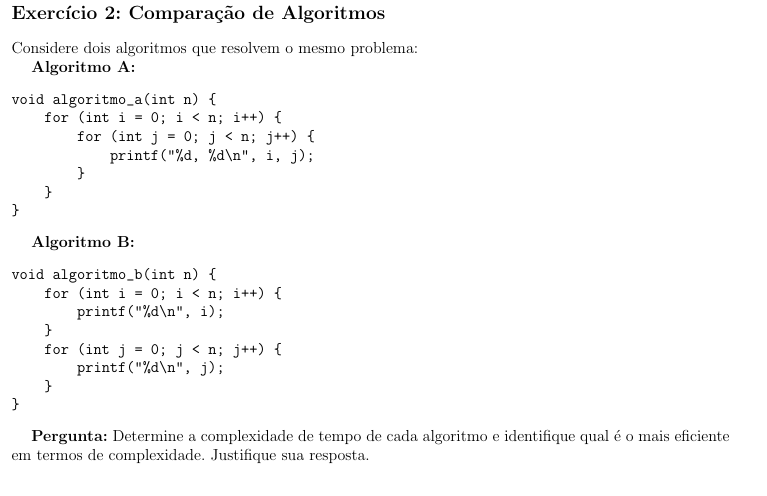
T(n) = n\*a + b

**Notação Big – O**

T(n) = a\*n Remover Constantes

T(n) = n Remover Constantes Multiplicativas

T(n) = O(n)



Resposta Questão 2 A:

C1\*n + C2 \* n2 + C3 \* n2

n2(C2 + C3) + C1\*n

n2a + n\*b

**Notação Big – O:**

T(n) = n2a + nb Remover Constantes

T(n) = n2 + n Remover Constantes Multiplicativas

T(n) = n2

**T(n) = O(n2)**

Resposta Questão 2B:

C1(n) + C2(n)

n(C1 + C2)

n\*a

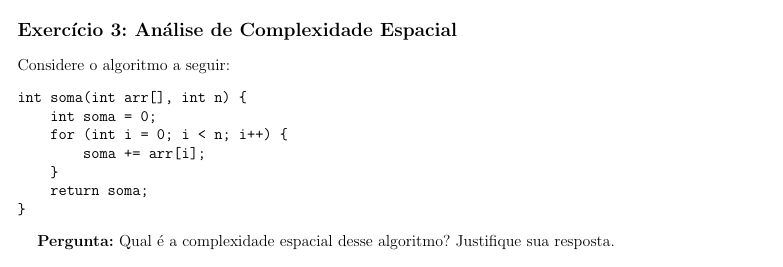
Notação **Big – O:**

T(n) = na Remover Constantes Multiplicativas

T(n) = n

**T(n) = O(n)**

**O Algoritmo B é mais eficiente em comparação com o Algoritmo A, pois a complexidade de tempo cresce de forma linear, já o Algoritmo A é uma função quadrática.**



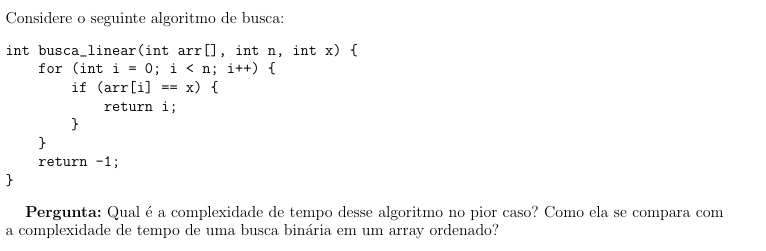
Resposta Questão 3:

O Espaço para arr[] é nB.

O espaço para n é 1B, para soma é 1B, para i é 1B.

S(soma) = 3B + nB

**S(soma) = B(3+n)**



Resposta Questão 4:

C1 \* n + C2 \* n + C3\*n

n(C1+C2+C3)

Na

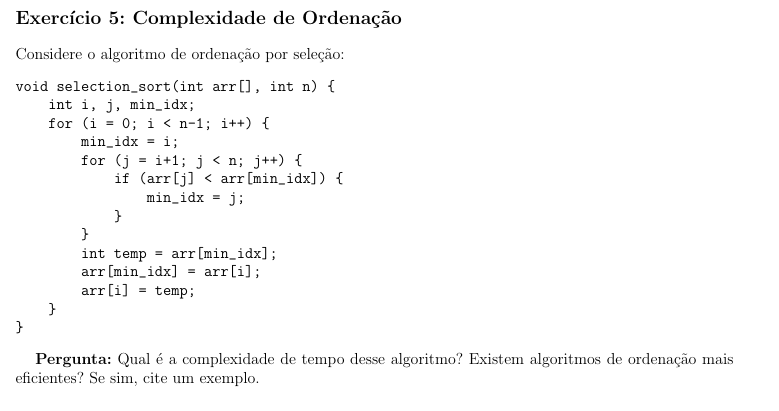
**Notação Big – O:**

T(n) = n\*a Remover Constantes Multiplicativas

T(n) = n

**T(n) = O(n).**

**A complexidade de uma busca binaria é (Log n), a função Logarítmica em questão de eficiência na complexidade de tempo comparado a função (n) linear se destaca com um melhor tempo de execução.**



Resposta Questão 5:

C1 + C2 \* n + C3 \* n+ C4 \* n2 + C5 \* n2 + C6 \* n2 + C7 \* n + C8 \* n + C9 \* n

N2(C4 + C5 + C6) + n(C2 + C3 + C7 + C8 + C9) + c

N2a + nb + c

**Notação Big – O:**

T(n) = n2a + nb + c Remover Constantes

T(n) = n2a + nb Remover Constantes Multiplicativas

T(n) = n2 + n Remover termos de grau menores.

T(n) = n2

**T(n) = O(n2).**

Sim. QuickSort com a complexidade (n Log n).