

Exercícios - Complexidade de Tempo de Algoritmos Iterativos

Análise de Algoritmos - DINF - UFPR

Para cada um dos algoritmos dos exercícios abaixo, pede-se:

- (a) Qual é o valor **exato** retornado pelo algoritmo? Expresse sua resposta em função de n .
- (b) Sabendo que a linha da operação $r \leftarrow r + 1$ é a mais executada, usando a notação O , analise o pior caso de execução do algoritmo.

Exercício 1.

Algoritmo *misterio*(n)

```
r ← 0
para i ← 1 até n faça
  r ← r + 1
retorna r
```

Exercício 2.

Algoritmo *misterio*(n)

```
r ← 0
para i ← 1 até n faça
  para j ← 1 até n faça
    r ← r + 1
retorna r
```

Exercício 3.

Algoritmo *misterio*(n)

```
r ← 0
para i ← 1 até n faça
  para j ← 1 até i faça
    r ← r + 1
retorna r
```

Exercício 4.

Algoritmo *misterio*(n)

```
r ← 0
para i ← 1 até n - 1 faça
  para j ← i + 1 até n faça
    para k ← 1 até j faça
      r ← r + 1
retorna r
```

Exercício 5.

Algoritmo *misterio*(n)

```
r ← 0
para i ← 1 até n faça
  para j ← 1 até i faça
    para k ← j até i + j faça
      r ← r + 1
retorna r
```

Exercício 6.**Algoritmo** *misterio*(n)

```

   $r \leftarrow 0$ 
  para  $i \leftarrow 1$  até  $n$  faça
    para  $j \leftarrow 1$  até  $i$  faça
      para  $k \leftarrow j$  até  $i + j$  faça
        para  $l \leftarrow 1$  até  $i + j - k$  faça
           $r \leftarrow r + 1$ 
  retorna  $r$ 

```

Para cada um dos algoritmos abaixo, pede-se:

- Identifique a **operação fundamental** do algoritmo (podem ser qualquer coisa dentre somas, multiplicações, ifs, atribuições, etc.).
- Analise o **número exato** de operações fundamentais executadas.
- Analise o pior caso do algoritmo usando a notação O .

Exercício 7.**Entrada:** vetor $A[1..n]$ **Saída:** o maior elemento de A

```

1 início
2    $m \leftarrow A[1]$ 
3   para  $i \leftarrow 2$  até  $n$  faça
4     se  $A[i] > m$  então  $m \leftarrow A[i]$ 
5   retorna  $m$ 

```

Exercício 8. Suponha constante $c \geq 2$:**Algoritmo** *multiplica*(y, z)

```

   $x \leftarrow 0$ 
  enquanto  $z > 0$  faça
     $x \leftarrow x + y \cdot (z \bmod c)$ 
     $y \leftarrow cy$ 
     $z \leftarrow \lfloor z/c \rfloor$ 
  retorna  $x$ 

```

Exercício 9.**Algoritmo** *power*(y, z)

```

   $x \leftarrow 1$ 
  enquanto  $z > 0$  faça
     $x \leftarrow x \cdot y$ 
     $z \leftarrow z - 1$ 
  retorna  $x$ 

```

Exercício 10.**Entrada:** vetor $A[1..n]$ **Saída:** o vetor A ordenado

```

1 início
2   para  $i \leftarrow 1$  até  $n - 1$  faça
3     para  $j \leftarrow 1$  até  $n - i$  faça
4       se  $A[j] > A[j + 1]$  então
5         Troque  $A[j]$  com  $A[j + 1]$ 
6   retorna  $A$ 

```