Universidade Federal de Ouro Preto
PCC104 - Projeto e Análise de Algoritmos
De volta ao básico

Prof. Rodrigo Silva

November 30, 2023

Questões

1. Resolva as seguintes relações de recorrência
   (a) $T(n) = T(n-1) + 5$, $T(1) = 0$
   (b) $T(n) = 3T(n-1)$, $T(1) = 4$
   (c) $T(n) = T(n-1) + n$, $T(0) = 0$
   (d) $T(n) = T(n/2) + n$, $T(1) = 1$
   (e) $T(n) = T(n/3) + 1$, $T(1) = 1$
2. Escreva um algoritmo recursivo que tenha custo definido por $T(n) = T(n-1) + 5$, $T(1) = 1$.
3. Escreva um algoritmo recursivo que tenha custo definido por $T(n) = 2T(n/2) + 1$, $T(1) = 1$.
4. Escreva um algoritmo iterativo que tenha complexidade $O(n^2)$.
5. Para cada um dos algoritmos abaixo, indique:
   (a) O que este algoritmo computa?
   (b) Qual a operação básica deste algoritmo?
   (c) Quantas vezes esta operação básica é executada?
   (d) Qual a classe deste algoritmo em relaçãoo à eficiência?

```python
def algorithm(x):
    result = 0
    for i in x:
        for j in x:
            for k in x:
                result += i * j * k
    return result
```

Figure 1: Algoritmo 1

```python
def algorithm(x):
    result = 0
    for i in x:
        result += i
    return result
```

Figure 2: Algoritmo 2

```python
def algorithm(x):
    result = 0
    for i in range(len(x)):
        for j in range(2,len(x)):
            result += i + j
    return result
```

Figure 3: Algoritmo 3

```python
def recursive_algorithm(n):
    if n == 1:
        return 0
    else:
        return recursive_algorithm(n - 1) + 5
```

Figure 4: Algoritmo 4

```python
def algorithm(x):
    result = 0
    for i in x:
        for j in x:
            for k in x:
                for l in x:
                    result += i * j * k * l
    return result
```

Figure 5: Algoritmo 5

```python
def recursive_algorithm(n):
    if n == 1:
        return 1
    else:
        val = 0
        for i in range(3):
            val += recursive_algorithm(n - 1)

        return val
```

Figure 6: Algoritmo 6

```python
def algorithm(n):
    if n <= 1:
        return n
    else:
        a = 0
        for i in range(n):
            a += i

        return a + algorithm(n // 2)
```

Figure 7: Algoritmo 7

```python
def algorithm_logn_cost(n):
    if n <= 1:
        return n
    else:
        return algorithm_logn_cost(n // 2) + 1
```

Figure 8: Algoritmo 8