Universidade Federal de Ouro Preto
PCC104 - Projeto e Análise de Algoritmos
Prova Especial

Prof. Rodrigo Silva

December 20, 2023

Questões

1. Apresente a análise de complexidade completa dos algoritmos abaixo.

   (a) Algoritmo 1

```
def useless_algorithm(r):

    for i in range(0, r):
        for j in range(0, i + 1):
            print("*", end=' ')
        print("\r")

    for i in range(r, 0, -1):
        for j in range(0, i - 1):
            print("*", end=' ')
        print("\r")
```

   (b) Algoritmo 2

```
def useless_algorithm2(n):

    for i in range(0,n):
        for j in range(i,n-1):
            print(j,end=' ')
        print()
```

   (c) Algoritmo 3

```
def is_upper_triangular(matrix[0,...,n-1][0,...,n-1]):

    for i in range(0,n):
        for j in range(i,n):
            if matrix[i][j] != 0:
                return False
    return True
```

(d) Algoritmo 4

```
def useless_algorithm3(a,b,c):

    for i in range(0,a):
        for j in range(0,b):
            print(j)
            for k in range(0,c):
                print(k)
        print()
```

(e) Algoritmo 5

```
def geometric_sum(n):
  if n < 0:
    return 0
  else:
    return 1 / (pow(2, n)) + geometric_sum(n - 1)
```

(f) Algoritmo 6

```
def print_pares(n):

    if n == 1:
        return

    if n % 2 == 0:
        print(n)

    print_pares(n-1)
```

(g) Algoritmo 7

```
def function(C, sol=[]):

    if len(sol) == len(C):
        print(sol)
    else:
        for v in range(len(C)):
            function(C, sol + [v])
```

(h) Algoritmo 8

```
def algoritmo(lista, inicio, fim):
    if fim - inicio <= 1:
        return lista[inicio]

    terco = (fim - inicio) // 3
    max1 = algoritmo(lista, inicio, inicio + terco)
    max2 = algoritmo(lista, inicio + terco, inicio + 2 * terco)
    max3 = algoritmo(lista, inicio + 2 * terco, fim)

    return max(max1, max2, max3)
```

(i) Algoritmo 9

```python
def algoritmo(lista, item, inicio=0, fim=None):
    if fim is None:
        fim = len(lista) - 1

    if inicio <= fim:
        meio = (inicio + fim) // 2
        chute = lista[meio]

        if chute == item:
            return meio
        if chute > item:
            return algoritmo(lista, item, inicio, meio - 1)
        else:
            return algoritmo(lista, item, meio + 1, fim)

    return None
```

(j) Algoritmo 10

```python
def find_max(matrix, row=0, col=0, max_val=None):

    n = len(matrix)
    m = len(matrix[0])

    if row == n:
        return max_val

    if matrix[row][col] > max_val:
        max_val = matrix[row][col]

    if col == m - 1:
        return find_max(matrix, row + 1, 0, max_val)
    else:
        return find_max(matrix, row, col + 1, max_val)
```