

Universidade Federal de Ouro Preto
PCC104 - Projeto e Análise de Algoritmos
Prova 6

Prof. Rodrigo Silva

December 19, 2023

Questões

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

(a) Algoritmo 1

```
1 for i in range(0, r):
2     for j in range(0, i + 1):
3         print("*", end=' ')
4     print("\r")
5
6 for i in range(r, 0, -1):
7     for j in range(0, i - 1):
8         print("*", end=' ')
9     print("\r")
```

(b) Algoritmo 2

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

(c) Algoritmo 3

```
1 def function(C, sol=[]):
2
3     if len(sol) == len(C):
4         print(sol)
5     else:
6         for v in range(len(C)):
7             function(C, sol + [v])
```

(d) Algoritmo 4

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

2. Faça a análise dos resultados obtidos por você com as implementações do Branch and Bound e da Busca Exaustiva para o problema do caixeiro viajante. Demonstre conhecimento sobre os algoritmos implementados e sobre a complexidade de cada um.
3. Dado um problema NP-completo, P_{NPC} , e um problema, $P_?$, que se deseja provar NP-completo responda:
 - (a) Quais são os dois passos para se provar que $P_?$ é NP-completo?
 - (b) Como a prova de cada passo pode ser feita?
 - (c) Por quê os dois passos são necessários? Seria possível determinar a NP-Completeness de $P_?$ só com o passo 1 ou só com o passo 2?
 - (d) Imagine que P_{NPC} foi resolvido por algum algoritmo em tempo polinomial. Qual seria o custo de se resolver $P2_{NPC}$ que sabemos ser NP-completo?