

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

Prof. Rodrigo Silva

July 12, 2023

Orientações

- É obrigatória a entrega do código fonte dos algoritmos gulosos. Provas sem os códigos fonte não serão corrigidas e terão nota 0.
- A avaliação do código apresentado entra na avaliação das questões relacionadas.

Questões

1. Análise de Algoritmo Iterativo Simples

Considerando o algoritmo de ordenação abaixo, que implementa o método de "Bubble Sort":

```
1 def bubble_sort(lista):
2     for i in range(len(lista)):
3         for j in range(0, len(lista) - i - 1):
4             if lista[j] > lista[j + 1]:
5                 lista[j], lista[j + 1] = lista[j + 1], lista[j]
```

- b) Qual é a complexidade de tempo no pior caso deste algoritmo?
- c) Qual é a complexidade de espaço deste algoritmo?

2. Análise de Algoritmo Recursivo Simples

Veja o seguinte algoritmo de busca binária:

```
1 def binary_search(array, low, high, target):
2     if high >= low:
3         mid = (high + low) // 2
4         if array[mid] == target:
5             return mid
6         elif array[mid] > target:
7             return binary_search(array, low, mid - 1, target)
8         else:
9             return binary_search(array, mid + 1, high, target)
10    else:
11        return -1
```

- a) Explique como este algoritmo funciona.
- b) Determine e justifique a complexidade de tempo no pior caso deste algoritmo.
- c) Determine e justifique a complexidade de tempo no melhor caso deste algoritmo.

3. Análise e Perguntas Teóricas Sobre o Branch and Bound

- a) Explique o conceito de branch and bound e sua aplicação na resolução de problemas de otimização.
- b) Compare o branch and bound com o método de força bruta. Em quais cenários cada um seria preferível e por quê?

- c) Como a estratégia de branch and bound pode impactar o custo computacional da resolução de um problema?
 - d) Apresente a análise de custo da sua implementação do branch and bound.
4. Perguntas Teóricas Sobre Classes de Problemas (P, NP, NP-completo)
- a) Defina as classes de problemas P, NP e NP-Completo.
 - c) É possível que $P = NP$? Explique sua resposta, discutindo as implicações se P fosse de fato igual a NP.