

Nome: _____

RA: _____ Assinatura: _____

Observações:

- Leia com cuidado a pergunta antes de começar. Você pode solicitar esclarecimentos durante o teste.
- Explique e justifique a sua resposta.
- Você também pode obter pontos por uma resposta parcial.
- É proibido qualquer tipo de consulta bibliográfica.
- O teste é individual, qualquer detecção de fraude implicará em zerar a nota da disciplina.

Questão 1 Considere o seguinte algoritmo para encontrar os $k \geq 1$ menores elementos de um vetor A com $n \geq k$ elementos:

Algoritmo 1: Menores(A, n, k)

```
1  $A[1 \dots k] \leftarrow$  Constrói uma max-heap com os primeiros  $k$  elementos de  $A$ 
2 para  $i \leftarrow k + 1$  até  $n$ 
3   se  $A[i] < A[1]$ 
4      $A[i] \leftrightarrow A[1]$ 
5     MaxHeapify( $A, k, 1$ )
6 devolva  $A[1 \dots k]$ 
```

- (a) **(3 pontos)** Qual a complexidade no melhor e pior caso do algoritmo em função de k e n ? Justifique.
- (b) **(5 pontos)** Considere uma versão aleatorizada do algoritmo acima: Adicione, no começo do algoritmo, uma linha que realiza uma permutação dos elementos de A , onde cada permutação tem igual chance de aparecer. Calcule a complexidade esperada dessa versão. As seguintes perguntas podem dar algumas dicas:
- O quê deve acontecer para $A[i]$ entrar na heap? É possível ter uma variável indicadora desse evento?
 - Qual a probabilidade desse evento? Pode ajudar calcular de quantas formas selecionar um dos k coelhos que mais pulam de um conjunto de i coelhos?
 - A partir do item anterior, é possível determinar quantas vezes MaxHeapify será chamada?
- (c) **(2 pontos)** Se $k \in O(n^\epsilon)$, para $\epsilon < 1$, qual a complexidade esperada do algoritmo aleatorizado? Justifique.