

**MAC0338 – Análise de Algoritmos**  
SEGUNDO SEMESTRE DE 2018  
Primeira Prova – 11 de setembro

Nome do aluno: \_\_\_\_\_ Curso: \_\_\_\_\_

Assinatura: \_\_\_\_\_

No. USP: \_\_\_\_\_ Professor: \_\_\_\_\_

### Instruções

1. Não destaque as folhas deste caderno.
2. A prova pode ser feita a lápis.
3. A legibilidade também faz parte da nota!
4. A prova consta de cinco questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é necessário apagar rascunhos no caderno de questão mas especifique qual é a resposta e qual é o rascunho.
7. A prova é sem consulta e tem duração de duas horas.
8. Você pode usar, sem ter que escrever, algoritmos vistos em aula. Neste caso, antes de usá-lo, deixe claro qual é o protótipo do algoritmo, o que ele devolve/faz, e quanto tempo ele consome em função de sua entrada.

**Não escrever nesta parte da folha**

Questão	Vale	Nota
1	1,5	
2	2,0	
3	2,0	
4	2,0	
5	2,5	
Total		

**Boa prova!**

1. [1,5 ponto]

Defina a função  $f: \mathbb{N} \rightarrow \mathbb{R}$  como  $f(n) := \sum_{i=1}^n i^2$  para todo  $n \in \mathbb{N}$ . Prove que  $f(n) = \Theta(n^3)$  **utilizando a definição de  $\Theta(\cdot)$** .

*Observação:* se você conhece uma fórmula fechada para  $f(n)$ , por exemplo um polinômio  $p(n)$  em  $n$  tal que  $f(n) = p(n)$  para todo  $n$ , você pode usar essa fórmula, *contanto que você a prove*. Se você não conhece tal fórmula, não se preocupe; é possível resolver a questão por outros meios.



2. [2,0 pontos]

Considere a recorrência

$$T(n) = T(n/3) + T(2n/3) + \Theta(n),$$

que descreve o consumo de tempo de um algoritmo. Estime a classe  $\Theta(\cdot)$  à qual  $T(n)$  pertence. Você pode usar tanto a receita vista em classe, expandindo a fórmula em diversas rodadas (para valores restritos de  $n$ , por exemplo potências de algum número), ou utilizar uma árvore de recursão. Em qualquer um dos casos, justifique bem sua resposta.



3. [2,0 pontos]

Descreva um algoritmo que recebe um vetor  $A[1..n]$  de  $n$  inteiros distintos e um inteiro  $x$  e determina se existem dois elementos de  $A$  cuja soma é  $x$ . Ou seja, o algoritmo determina se existem índices  $i, j \in [n]$  tais que  $A[i] + A[j] = x$ .

**O consumo de tempo do seu algoritmo deve ser  $O(n \lg n)$ .**

Justifique a corretude e o consumo de tempo do seu algoritmo. Não é necessário escrever um pseudocódigo se sua descrição estiver precisa o suficiente.



4. [2,0 pontos]

Considere o algoritmo INSERTIONSORT( $A, n$ ) visto em sala e reproduzido abaixo.

INSERTIONSORT( $A, n$ )

1. **para**  $j \leftarrow 2$  **até**  $n$  **faça**
2.      $chave \leftarrow A[j]$
3.      $i \leftarrow j - 1$
4.     **enquanto**  $i \geq 1$  e  $A[i] > chave$  **faça**
5.          $A[i + 1] \leftarrow A[i]$
6.          $i \leftarrow i - 1$
7.      $A[i + 1] \leftarrow chave$

Suponha que  $A[1..n]$  é uma permutação de 1 a  $n$  escolhida ao acaso dentre todas as permutações de 1 a  $n$ , de acordo com a distribuição uniforme de probabilidade. Seja  $X$  o número de vezes que a linha 5 do algoritmo é executada. Note que  $X$  é uma variável aleatória. Calcule o valor esperado de  $X$ . Justifique a sua resposta.

*Dica:* Lembre-se que, num vetor  $A[1..n]$ , um par de índices  $(i, j)$  com  $i, j \in [n]$  é uma *inversão* se  $i < j$  e  $A[i] > A[j]$ .



5. [2,5 pontos]

Escreva o pseudocódigo para um algoritmo  $\text{MEDIANA2}(A, B, n)$  que recebe dois vetores ordenados (em ordem crescente)  $A[1..n]$  e  $B[1..n]$ , cada um contendo  $n$  inteiros, e devolve uma das duas medianas desses  $2n$  inteiros. (Lembre-se que a mediana de  $m$  inteiros é o  $\lfloor \frac{m+1}{2} \rfloor$ -ésimo menor ou o  $\lceil \frac{m+1}{2} \rceil$ -ésimo menor desses  $m$  inteiros.)

**O consumo de tempo do seu algoritmo deve ser  $O(\lg n)$ .**

Justifique a corretude e o consumo de tempo do seu algoritmo.

*Dica:* Escreva uma função auxiliar  $\text{AUX2}(A, p_A, r_A, B, p_B, r_B, k)$  que opera sobre os subvetores  $A[p_A..r_A]$  e  $B[p_B..r_B]$ .

