



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
MINAS GERAIS  
Campus Formiga

**Graduação em Ciência da Computação**

**Disciplina: Projeto e Análise de Algoritmos**

**Professor: Wallace de Almeida Rodrigues**

**Atividade: 2a prova – 01/abr/2021**

**Ensino Emergencial Remoto**

**Código de honra e conduta discente:**

Eu, \_\_\_\_\_, matrícula \_\_\_\_\_, prometo pela minha honra que fui honesto e não trapaceei nessa avaliação passando ou recebendo cola.

**Formiga, MG, 1 de abril de 2021.**

---

### **INSTRUÇÕES:**

1. Siga as instruções, a alegação de desconhecimento das regras não será considerada.
  2. Esta prova deve, obrigatoriamente, ser resolvida individualmente. Resolva a prova à mão e grafite suas respostas de forma perfeitamente legível. A organização e legibilidade da solução enviada será avaliada, portanto cuide da qualidade da apresentação.
  3. A solução da prova deve ser entregue, obrigatoriamente, em um único arquivo no formato PDF.
  4. Caso você ache que falta algum detalhe nas especificações, faça as suposições que julgar necessárias e as escreva junto à resolução do trabalho. Pode acontecer também que a descrição dessa atividade contenha dados e/ou especificações supérfluas para sua solução. Utilize sua capacidade de julgamento para separar o supérfluo do necessário.
  5. No início da solução deve constar esta página contendo o código de honra e conduta preenchido e assinado.
  6. Nas questões, não basta escrever apenas o resultado final. É necessário mostrar o raciocínio utilizado para responder às questões. Por se tratar de uma prova, as questões foram elaboradas para avaliar conhecimento e competência nos temas abordados, portanto responda as questões de forma clara e completa o bastante para demonstrar que você entendeu bem os conceitos envolvidos na disciplina.
  7. Ao construir o único arquivo PDF, cuide para que sua resposta fique organizada e legível!
  8. A solução desta prova deve ser entregue, obrigatoriamente, via página da disciplina no Google Classroom. Entregue um único arquivo no formato PDF.
  9. O prazo final para entrega desta atividade é até 23:59 do dia 1/04/2021. O envio é de total responsabilidade do aluno. Soluções enviadas fora do prazo estabelecido serão muito penalizadas.
  10. Resoluções plagiadas serão desconsideradas, sendo atribuída nota 0 (zero) a todos os envolvidos.
  11. O valor desta atividade é 25 pontos.
-

**Questão 1.** (5 ptos) – A recorrência  $A(n) = 5A(n/2) + n^2$  descreve o tempo de execução de um algoritmo A. Um algoritmo alternativo B tem um tempo de execução  $B(n) = kB(n/8) + n^2$ . Qual é o maior número inteiro  $k$  que faz  $B$  ser assintoticamente mais rápido que  $A$ ? Justifique sua resposta.

---

**Questão 2.** (8 ptos) – Em cada caso abaixo, uma função  $T(n)$  é apresentada de forma recursiva. Encontre, em cada caso, uma função  $H(n)$ , apresentada de forma não recursiva, tal que  $T(n) \in \Theta(H(n))$ . Você precisa justificar sua resposta de algum modo (por exemplo, recorrendo adequadamente ao Teorema Mestre apontando o caso aplicado e porque se aplica, fazendo análise do custo resolvendo a recorrência, etc). Assuma em todos os casos que  $T(1)$  é uma constante positiva.

- (a)  $T(n) = 10T(\lfloor n/2 \rfloor) + n^3$
  - (b)  $T(n) = T(n-1) + 5$
  - (c)  $T(n) = 2T(n/2) + n * \log n$
  - (d)  $T(n) = 2T(\lfloor n/4 \rfloor) + T(\lfloor n/2 \rfloor)$
- 

**Questão 3.** (12 ptos) – Analise os códigos python abaixo e calcule a **complexidade assintótica** em cada caso usando a notação *Big O*. Na sua resposta, considere a semântica envolvida nas instruções complexas, levando em conta seus conhecimentos de arquitetura de computadores, tradução de instruções e programação de baixo nível. Justifique suas respostas de acordo com esses padrões apontando a interpretação adotada. Considere o pior caso.

DICA: na sua resposta indique o custo de cada linha do código.

Algoritmo A

```
1 soma = 0;
2 for i in range(n):
3     for j in range(int(n/3)):
4         k = 1;
5         while (k <= n):
6             soma = soma + 1;
7             k = k * 2;
```

Algoritmo B

```
1 n = random.choice(range(10))
2 lista = random.sample(range(100), n)
3 x = random.choice(lista)
4 for i in lista[:100]:
5     if (i < x):
6         escolhidos.append(x*2);
7     else:
8         for j in range(i):
9             escolhidos.append(i);
```

### Algoritmo C

```
1 def M(n):  
2     soma = 0;  
3     if (n > 2):  
4         for i in range(n):  
5             soma = M(n/2) + 1;  
6     return soma;
```

### Algoritmo D

```
1 def T(n):  
2     soma = 0;  
3     while (n > 1):  
4         n = int(n / 2);  
5         soma = T(n) + 1;  
6     return soma;
```

---

***Boa Prova!***