

**QUESTÃO 1** (3 pontos)

Considere que a multiplicação de duas matrizes é  $O(n^3)$ . Se você tivesse a opção de utilizar outro algoritmo da ordem  $O(2^n)$  para multiplicar duas matrizes, qual algoritmo você preferiria? Justifique

**QUESTÃO 2** (6 pontos)

Resolva a equação de recorrência encontrando sua forma fechada

$$\begin{cases} T(n) = 3T(n/3) + \frac{5}{3}n - 2 & \text{para } n > 1 \\ T(1) = 0 \end{cases}$$

**QUESTÃO 3** (5 pontos)

Encontre a função de complexidade do algoritmo abaixo em relação ao número de comparações.

```
1 void Proc(vetor a, tamanho n){
2     for (int j=2; j<=n-3; j++) {
3         for (i=n; i>i; i--) {
4             if (a[i-2] < a[i-1]) {
5                 troca(a[i-2], a[i-1]);
6             }
7         }
8     }
9 }
```

**QUESTÃO 4** (6 pontos)

Segundo o professa Rutyle, uma das maneiras de se conseguir a raiz quadrada de um número é subtrair deste número os ímpares consecutivos a partir de 1, até que o resultado da subtração seja menor ou igual a zero. O número de vezes que se conseguir fazer a subtração é a raiz quadrada exata (resultado 0) ou aproximada do número (resultado negativo). Exemplo, para encontrar a raiz de 16:

16 - 1  
15 - 3  
12 - 5  
7 - 7

Assim, a raiz de 16 é 4, pois foram necessárias 4 subtrações até chegar a zero. Faça um algoritmo RECURSIVO para provar se a tese da professora está correta ou não.

BOA PROVA!