

Lista 4 — MC458

Tema: Corretude e Teorema Mestre

1. (**Problema 4.1 — Tim Roughgarden**)

Relembre o Método Mestre (Teorema 4.1 do livro de Tim Roughgarden) e seus três parâmetros a , b e d . Qual das seguintes opções é a melhor interpretação de b^d ?

- a) A taxa na qual o trabalho total cresce (por nível de recursão).
- b) A taxa na qual o número de subproblemas cresce (por nível de recursão).
- c) A taxa na qual o tamanho do subproblema encolhe (por nível de recursão).
- d) A taxa na qual o trabalho por subproblema encolhe (por nível de recursão).

2. (**Problema 4.2 — Tim Roughgarden**)

Suponha que o tempo de execução $T(n)$ de um algoritmo seja limitado por uma recorrência padrão com $T(n) \leq 7 \cdot T(n/3) + O(n^2)$. Qual das seguintes opções é o menor limite superior correto para o tempo de execução assintótico do algoritmo?

- a) $O(n \log n)$
- b) $O(n^2)$
- c) $O(n^2 \log n)$
- d) $O(n^{2.81})$

3. (**Problema 4.3 — Tim Roughgarden**)

Suponha que o tempo de execução $T(n)$ de um algoritmo seja limitado por uma recorrência padrão com $T(n) \leq 9 \cdot T(n/3) + O(n^2)$. Qual das seguintes opções é o menor limite superior correto para o tempo de execução assintótico do algoritmo?

- a) $O(n \log n)$
- b) $O(n^2)$
- c) $O(n^2 \log n)$
- d) $O(n^{3.17})$

4. (**Problema 4.4 — Tim Roughgarden**)

Suponha que o tempo de execução $T(n)$ de um algoritmo seja limitado por uma recorrência padrão com $T(n) \leq 5 \cdot T(n/3) + O(n)$. Qual das seguintes opções é o menor limite superior correto para o tempo de execução assintótico do algoritmo?

- a) $O(n^{\log_5 3})$
- b) $O(n \log n)$
- c) $O(n^{\log_3 5})$
- d) $O(n^{5/3})$
- e) $O(n^2)$
- f) $O(n^{2.59})$

5. Análise de Algoritmos Reais (Karatsuba e Strassen)

Dadas as equações de recorrência de dois algoritmos clássicos de Divisão e Conquista:

- a) **Multiplicação de Karatsuba:** $T(n) \leq 3T(n/2) + O(n)$. Determine a complexidade assintótica em notação Big- O .
- b) **Multiplicação de Matrizes de Strassen:** $T(n) \leq 7T(n/2) + O(n^2)$. Determine a complexidade assintótica em notação Big- O .
- c) Comparando o algoritmo de Strassen com a abordagem tradicional de Divisão e Conquista ($T(n) = 8T(n/2) + O(n^2)$), explique por que o algoritmo de Strassen é assintoticamente mais rápido com base no impacto do parâmetro a em relação a b^d .

6. Subvetor de Soma Máxima (Divisão e Conquista)

Considere o problema de encontrar a maior soma contínua em um vetor $A[1 \dots n]$ de inteiros resolvido por Divisão e Conquista:

```
MaxSubarray(A, low, high):
    se low == high então
        retorne max(0, A[low])

    mid = ⌊(low + high) / 2⌋
    max_esq = MaxSubarray(A, low, mid)
    max_dir = MaxSubarray(A, mid + 1, high)
    max_cruz = MaxCruzado(A, low, mid, high)

    retorne max(max_esq, max_dir, max_cruz)

MaxCruzado(A, low, mid, high):
    soma = 0,    soma_esq =  $-\infty$ 
    para  $i$  de mid passando até low passo -1 faça
        soma = soma + A[i]
        se soma > soma_esq então soma_esq = soma

    soma = 0,    soma_dir =  $-\infty$ 
    para  $j$  de mid + 1 até high faça
        soma = soma + A[j]
        se soma > soma_dir então soma_dir = soma

    retorne soma_esq + soma_dir
```

- a) Prove a corretude do algoritmo em duas etapas:
 - i) Prove que a função **MaxCruzado** devolve a maior soma entre todos os subvetores contínuos que cruzam o ponto médio mid (isto é, $A[i \dots j]$ com $low \leq i \leq mid < j \leq high$).
 - ii) Prove por **Indução Matemática** no tamanho do vetor $n = high - low + 1$ que a função principal **MaxSubarray** retorna a soma máxima de qualquer subvetor contínuo em $A[low \dots high]$.

- b) Escreva a equação de recorrência $T(n)$ do tempo de execução do algoritmo. Em seguida, identifique os parâmetros a , b e d do Teorema Mestre de Tim Roughgarden e determine a complexidade assintótica final em notação Big- O .

7. Busca por Elemento Majoritário

Um elemento x é dito **majoritário** em um vetor $A[1 \dots n]$ se sua frequência é estritamente maior que $n/2$. Considere o algoritmo por Divisão e Conquista abaixo:

```
Majoritario(A, low, high):
    se low == high então
        retorne A[low]

    mid = ⌊(low + high) / 2⌋
    cand_esq = Majoritario(A, low, mid)
    cand_dir = Majoritario(A, mid + 1, high)

    se cand_esq == cand_dir então
        retorne cand_esq

    freq_esq = Contar(A, low, high, cand_esq)           // 0(n)
    freq_dir = Contar(A, low, high, cand_dir)           // 0(n)

    se freq_esq > (high - low + 1) / 2 então retorne cand_esq
    se freq_dir > (high - low + 1) / 2 então retorne cand_dir
    retorne NENHUM
```

- a) Prove a seguinte propriedade essencial para a corretude do algoritmo: *Se um elemento x é majoritário no vetor completo $A[1 \dots n]$, então x obrigatoriamente deve ser o elemento majoritário na metade esquerda ou na metade direita do vetor.*
- b) Formule a equação de recorrência $T(n)$ para o pior caso. Identifique os parâmetros a , b e d , interprete a relação entre eles e determine a complexidade assintótica em notação Big- O .

8. Localização de Pico em Vetor (Peak Finding)

Um elemento $A[i]$ em um vetor $A[1 \dots n]$ é chamado de **pico** se $A[i] \geq A[i - 1]$ e $A[i] \geq A[i + 1]$ (considere $A[0] = A[n + 1] = -\infty$). Considere o algoritmo:

```
EncontraPico(A, low, high):
    mid = ⌊(low + high) / 2⌋

    se A[mid] < A[mid + 1] então
        retorne EncontraPico(A, mid + 1, high)
    senão se A[mid] < A[mid - 1] então
        retorne EncontraPico(A, low, mid - 1)
    senão
        retorne mid
```

- a) Prove por **Indução** que o subvetor $A[\text{low} \dots \text{high}]$ sempre contém pelo menos um pico, garantindo que o algoritmo sempre retorna um pico válido.
- b) Monte a recorrência de tempo $T(n)$, identifique os parâmetros a , b e d do Teorema Mestre de Tim Roughgarden e calcule a complexidade assintótica do algoritmo.