Lista de exercícios 2
Cana - 2025.1

Questão 1. Resolva as seguintes relações de recorrência (com pelo menos uma solução diferente do método mestre, para cada item):

(a) $T(n) = T(n-1) + n$;

(b) $T(n) = T(n/2) + n$;

(c) $T(n) = T(n/5) + T(7n/10) + n$;

(d) $T(n) = 4T(n/2) + n$;

(e) $T(n) = 3T(n/2) + n^2$;

(f) $T(n) = T(n/3) + T(2n/3) + n$;

Questão 2. Considere o eguinte algoritmo de ordenação:

Algoritmo 1: X(A, i, j)

```
1 se A[i] > A[j] então
2 |  Troca A[i] ↔ A[j]
3 se i + 1 ≥ j então
4 |  Retorna
5 k ← ⌊(j − i + 1)/3⌋
6 X(A, i, j − k)
7 X(A, i + k, j)
8 X(A, i, j − k)
```

Apresente uma recorrência para o pior caso de tempo de execução do algoritmo, mostrando um limite assintótico apertado para a mesma. Prove esse limite utilizando um método a sua escolha (exceto o método mestre). Esse algoritmo é o mais eficiente possível para a realização da tarefa? Explique por que.

Questão 3. Você está tentando escolher entre os três algoritmos A, B e C descritos a seguir. O **Algoritmo A** resolve o problema dividindo a entrada em cinco subproblemas com a metade do tamanho, resolve cada subproblema recursivamente e depois combina-os em tempo linear. O **Algoritmo B** resolve o problema dividindo a entrada em dois subproblemas de tamanho $n-1$ (onde n é o tamanho da entrada), resolve cada um recursivamente e combina-os em tempo constante. O **Algoritmo C** resolve o problema dividindo a entrada em nove subproblemas com um terço do tamanho, resolve cada subproblema recursivamente e depois combina-os em tempo quadrático. Qual o tempo de cada um em notação assintótica? Utilize um método diferente para cada algoritmo na realização desse cálculo. Qual você escolheria?

Questão 4. Faça um algoritmo de divisão e conquista para multiplicar duas matrizes quadradas de dimensões $n \times n$, dividindo cada matriz em 4 submatrizes quadradas de dimensões $\frac{n}{2} \times \frac{n}{2}$, assumindo que o número de linhas e colunas é uma potência de 2. Apresente uma recorrência para o tempo de execução do seu algoritmo .

Questão 5. Suponha que os pivots escolhidos em uma execução de QUICK-SORT particionam o vetor na proporção $9:1$. Calcule a complexidade do algoritmo neste caso.

Questão 6. Como você modificaria QUICKSORT para ordenar em ordem não crescente?

Questão 7. Apresente uma implementação do algoritmo COUNTING-SORT que não utiliza o vetor auxiliar B (com apenas um vetor auxiliar).

Questão 8. O algoritmo RADIX-SORT recebe como entrada números com vários dígitos (no máximo d) e os ordena a partir do dígito menos significativo. Apresente uma implementação deste algoritmo.

Questão 9. O algoritmo do k-ésimo mínimo elemento ainda seria $\Theta(n)$ se tomássemos grupos de 3 elementos, ao invés de 5? E se tomássemos grupos de 7 elementos? Justifique usando o método da árvore de recursão.

Questão 10 (3,5 pontos)**.** Apresente um algoritmo para ordenar um vetor de inteiros $A[1 \ldots n]$ em tempo $O(n + M)$, onde $M = max(A[i]) - min(A[j])$, para $1 \leq i, j \leq n$. Para M pequeno, isso é tempo linear: essa complexidade contradiz o limite inferior de ordenação baseado em comparações de $O(n\ log\ n)$?

Questão 11. Seja $X[1 \ldots n]$ um vetor de números reais. Dizemos que X tem um elemento *popular* x se mais de *um terço* de seus elementos são iguais a x. Escreva um algoritmo de tempo linear $\Theta(n)$ que diz se X possui ou não um elemento popular. Caso sim, devolva o seu valor.

Questão 12. Descreva um algoritmo que, dados n inteiros na faixa 0 a k, reprocesse sua entrada e depois responde a qualquer consulta sobre quantos dos n inteiros caem em uma faixa $[a..b]$ no tempo $O(1)$. Seu algoritmo deve utilizar o tempo de pré-processamento $O(n + k)$.

Questão 13. Mostre como ordenar n inteiros na faixa de 0 a $n^3 - 1$ no tempo $O(n)$.

Questão 14. Mostre que o segundo menor entre n elementos pode ser determinado com $n + log\ n - 2$ comparações no pior caso.

Questão 15. Descreva um algoritmo de tempo $O(n)$ que, dados um conjunto S de n números distintos e um inteiro positivo $k \leq n$, determine os k números em S que estão mais próximos da mediana de S.

Questão 16. Suponha que um algoritmo utilize somente comparações para determinar o i-ésimo menor elemento em um conjunto de n elementos. Mostre que ele também pode determinar os $i - 1$ menores elementos e os $n - i$ maiores elementos sem executar quaisquer comparações adicionais.