

Lista de exercícios 2
Cana - 2023.1

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. Considerado o algoritmo abaixo, seja $A(n)$ o valor retornado por `NUMRECURSIVO(N)`. Justifique suas respostas pelo **método da árvore de recursão**.

- (a) Escreva uma recorrência para o valor $A(n)$ retornado e calcule $A(n)$ em notação assintótica.
- (b) Troque “ $n/4$ ” por “ $n/5$ ” no algoritmo e calcule novamente $A(n)$.

Algoritmo 1: inteiro NUMRECURSIVO(inteiro N)

```
Valor = 0    //(Valor é uma variável local)
if  $N > 1$  then
    Valor = Valor + NUMRECURSIVO( $\lfloor N/2 \rfloor$ )
    Valor = Valor + NUMRECURSIVO( $\lfloor N/4 \rfloor$ ) + NUMRECURSIVO( $\lfloor N/4 \rfloor$ ) + N
end if
retorne Valor
```

Questão 3. 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 4. Apresente uma implementação do algoritmo COUNTING-SORT que não utiliza o vetor auxiliar B (com apenas um vetor auxiliar).

Questão 5. 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 6. Seja $X[1 \dots n]$ 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 7. Seja M uma matriz qualquer com n linhas e n colunas (os elementos dessa matriz não são necessariamente inteiros ou caracteres; podem ser objetos quaisquer, como frutas ou arquivos executáveis). Suponha que você possui apenas um operador “=” que permite comparar se um objeto é igual a outro. Dizemos que a matriz M tem um elemento super-majoritário x se mais de três quartos ($3/4$) de seus elementos são iguais a x . Descreva com palavras (e não utilizando pseudocódigo ou qualquer linguagem) um algoritmo de tempo $\Theta(n^2 \log n)$ que diz se a matriz M possui ou não um elemento super-majoritário. Caso sim, devolva o seu valor.

Questão 8. Dizemos que um algoritmo é de *quase-ordenação* se, para qualquer vetor $A[1 \dots n]$, o algoritmo rearranja os valores do vetor A de modo que $i < j$ implica $A[i] < A[j] + 0.1$. Por exemplo, o vetor $A = [1.5 \ 1.45 \ 2.4 \ 2.35 \ 3]$ está quase-ordenado. Sabendo que os valores do vetor de entrada são números reais menores que 100, faça um algoritmo de tempo $O(n)$ para quase-ordenar o vetor.