

### Modalidade:

- síncrono com os horários de aula

**Data de entrega:** 30/julho/2021 até 23:59 hs no ambiente Microsoft Teams.

**Tópico: Técnicas de Projetos de Algoritmos**

Em cada exercício calcule as complexidades de tempo e espaço dos algoritmos projetados.

1. Suponha que seja dado um algoritmo caixa-preta (CP) de complexidade  $f_{CP}(n)$  com a seguinte propriedade: dada uma sequência  $A$  de  $n$  números inteiros e um inteiro  $k$ , CP retorna verdadeiro ou falso, indicando se existe um subconjunto de números cuja soma é exatamente  $k$ . Usando o algoritmo CP, projete por indução um algoritmo que retorne os elementos do subconjunto cuja soma é  $k$ . Calcule a complexidade de seu algoritmo em função de  $f_{CP}(n)$ .

$CP(A, n, k)$ : retorna verdadeiro se  $A$  possui subconjunto cuja soma seja exatamente  $k$ .

2. Um encanador necessita fazer  $n$  reparos urgentes, e sabe de antemão o tempo que leva cada um deles: o reparo  $i$ -ésimo leva  $t_i$  minutos. Como em sua empresa lhe pagam dependendo da satisfação do cliente, necessita decidir a ordem que ele atenderá os pedidos para minimizar o tempo médio de espera dos clientes. Isto é, seja  $E_i$  o tempo de espera do  $i$ -ésimo cliente pela reparação completa da sua avaria. Então é preciso minimizar:

$$E(n) = \sum_{i=1}^n E_i$$

Projete um algoritmo guloso que resolva o problema.

3. Considere a função de Ackermann, definida pela recorrência abaixo:

$$\begin{cases} Ack(0, n) = n + 1 \\ Ack(m, 0) = Ack(m - 1, 1) \\ Ack(m, n) = Ack(m - 1, Ack(m, n - 1)) \quad \text{se } m, n > 0 \end{cases}$$

Projete um algoritmo de Programação Dinâmica para calcular  $Ack(m, n)$ .

4. (Problema do Troco). Considere um sistema monetário formado por moedas de valores  $v_1, v_2, \dots, v_n$ . O problema do troco consiste em decompor uma quantidade de dinheiro  $M$  em moedas utilizando o menor número possível de moedas. Suponha que há quantidade ilimitada de moedas de todos os valores.

Exemplo:  $M = 10$  e  $V = \{1, 5\}$ . Podemos usar:

- (a) 10 moedas de 1 centavo:  $10 \times 1 = 10$  (10 moedas usadas)
- (b) 1 moeda de 5 centavos, e 5 moedas de 1 centavo:  $1 \times 5 + 5 \times 1 = 10$  (6 moedas usadas)
- (c) 2 moedas de 5 centavos:  $2 \times 5 = 10$  (2 moedas usadas)  $\rightarrow$  **ótimo**

Projete um algoritmo para resolver o problema do troco.