

## ***Lista de exercícios 11 – tipos de algoritmos***

Exercício 1: Algoritmo Guloso (Problema da Troca de Moedas)

Implemente um programa que, dado um valor inteiro  $N$  e um conjunto de moedas de valores diferentes, encontre a menor quantidade de moedas necessárias para representar o valor  $N$ . Considere que há um número infinito de moedas de cada valor disponível.

*Exemplo:*

*Entrada:*

$N = 36$

*Conjunto de moedas:* [1, 5, 10, 25]

*Saída:*

*Quantidade mínima de moedas:* 3 (utilizando 1 moeda de 25, 1 moeda de 10 e 1 moeda de 1)

Exercício 2: Algoritmo de Divisão e Conquista (Ordenação por Fusão)

Implemente um programa que utilize o algoritmo de ordenação por fusão (merge sort) para ordenar uma lista de valores inteiros.

*Exemplo:*

*Entrada:*

*Lista não ordenada:* [7, 2, 9, 1, 6]

*Saída:*

*Lista ordenada:* [1, 2, 6, 7, 9]

**Exercício 3: Algoritmo Guloso (Problema da Mochila Fracionária)**

Implemente um programa que, dado um conjunto de objetos com pesos e valores, encontre a melhor combinação de objetos para colocar em uma mochila com uma capacidade máxima, maximizando o valor total. Neste caso, os objetos podem ser fracionados.

*Exemplo:*

*Entrada:*

*Capacidade da mochila: 50*

*Conjunto de objetos: [(60, 10), (100, 20), (120, 30)]*

*Saída:*

*Valor máximo alcançado: 240 (utilizando 20 unidades do objeto 1 e 30 unidades do objeto 3)*

**Exercício 4: Algoritmo de Divisão e Conquista (Pesquisa Binária)**

Implemente um programa que utilize o algoritmo de pesquisa binária para encontrar um elemento em uma lista ordenada de valores inteiros. O programa deve retornar a posição do elemento na lista ou -1 caso não seja encontrado.

*Exemplo:*

*Entrada:*

*Lista ordenada: [1, 4, 6, 9, 13, 20, 22]*

*Elemento a ser pesquisado: 9*

*Saída:*

*Posição do elemento: 3*