



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
**Faculdade de Ciências e Tecnologia - Câmpus
Presidente Prudente**

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
- FACULDADE DE CIÊNCIAS E TECNOLOGIA - CÂMPUS PRESIDENTE
PRUDENTE

IGOR MENDES DOMINGUES MIRAS
GUSTAVO SANTOS GONÇALVES

TRABALHO PRÁTICO II

PRESIDENTE PRUDENTE
2023

1. INTRODUÇÃO

O presente trabalho discorre sobre problemas clássicos da Ciência da Computação, envolvendo diferentes abordagens em termos de projeto de algoritmo. Dentre os vários problemas existentes nesse ramo da computação, os abordados neste trabalho são: Problema de Associação de Tarefas (Assignment Problem), Codificação de Huffman (Huffman Coding), Problema da Mochila Fracionária (Fractional Knapsack Problem), Problema da Mochila Booleana (Knapsack Problem) e Problema da Subsequência Comum Máxima (Longest Common Subsequence). As diferentes estratégias de projeto de algoritmos utilizadas para resolver os problemas citados acima são: Branch and Bound, Algoritmo Guloso e Programação Dinâmica.

2. PROBLEMA DE ASSOCIAÇÃO DE TAREFAS

Desafios relacionados à atribuição manifestam-se em diversas circunstâncias, seja na distribuição de responsabilidades entre membros da equipe, na alocação de motoristas para trajetos específicos, ou em uma variedade de outros contextos. O objetivo central da resolução deste problema consiste em identificar uma atribuição que otimize critérios específicos, como custo ou tempo.

Especificamente no caso do Problema de Associação de Tarefas, o problema consiste em atribuir N pessoas a N tarefas, sendo que cada pessoa possui um valor (podendo ser considerado como custo ou tempo) associado a cada tarefa.

A solução utilizada neste trabalho para a resolução do problema foi o uso de um algoritmo de Força Bruta - Branch and Bound, em que todas as possibilidades possíveis são testadas.

Complexidade: $O(n!)$, (sendo n o número de tarefas e também o número de pessoas).

	TAREFA 1	TAREFA 2	TAREFA 3
PESSOA 1	5	8	3
PESSOA 2	2	9	1
PESSOA 3	7	6	4

3. CODIFICAÇÃO DE HUFFMAN

A Codificação de Huffman é uma técnica geralmente utilizada para comprimir dados que consistem em caracteres frequentemente repetidos. Para a resolução desse problema, a abordagem mais interessante é a de um Algoritmo Guloso, que consiste em escolher um ótimo local a cada iteração, a fim de, no final das iterações, alcançar um ótimo global. A estrutura de dados árvore binária de busca foi utilizada nessa implementação, em que um nó é criado para representar cada caractere pertencente a frase digitada pelo usuário.

Complexidade: $O(n \log(n))$, (sendo n o tamanho da frase).

Considere, por exemplo, a entrada “paralelepipedo”. Cada caractere ficará da seguinte maneira de acordo com o algoritmo:

‘p’	01
‘e’	00
‘a’	101
‘l’	100
‘r’	1101
‘i’	1100
‘d’	1111
‘o’	1110

Por fim, após a Codificação de Huffman, a saída será **0110111011011000010000011100010011111110**.

4. PROBLEMA DA MOCHILA FRACIONÁRIA

Dado os pesos e lucros de N itens, na forma {lucro, peso}, coloque esses itens em uma mochila de capacidade W para obter o lucro total máximo na mochila. No Problema da Mochila Fracionária, o objetivo é saber que fração de cada objeto deve ser colocado na mochila de modo a maximizar o lucro total. Para a resolução desse problema, o ideal é a utilização de um Algoritmo Guloso (já definido anteriormente), buscando um ótimo local para atingir um ótimo global.

Complexidade: $O(n \log(n))$, (sendo n o tamanho do vetor de entrada).

Considerando, por exemplo, uma mochila com capacidade de 10Kg e a seguinte entrada:

	PESO	VALOR
ITEM 1	7	10
ITEM 2	2	1
ITEM 3	5	6

A saída será a seguinte:

	PORCENTAGEM
ITEM 1	100%
ITEM 2	60%
ITEM 3	0%

Portanto, o valor acumulado na mochila será de **13,60**.

5. PROBLEMA DA MOCHILA BOOLEANA

Dado N itens onde cada item possui um peso e lucro associados, e também dada uma mochila com capacidade W (ou seja, a mochila pode conter no máximo peso W), a tarefa é colocar os itens na mochila de forma que a soma dos lucros associados a eles seja a máxima possível. O presente problema se diferencia do problema anterior graças a sua abordagem binária, onde ou um item cabe completamente na mochila, ou então não cabe (booleano). Para solucionar esse problema, a abordagem mais adequada é a de um algoritmo baseado em Programação Dinâmica, que é uma técnica poderosa para resolver problemas de otimização, decompondo o problema em subproblemas menores, resolvendo-os de forma ótima e armazenando os resultados a fim de evitar cálculos redundantes, resultando em uma solução eficiente.

Complexidade: $O(n^2)$, (sendo n o tamanho do vetor de entrada).

Considerando, por exemplo, uma mochila com capacidade de 10Kg e a mesma entrada do problema anterior:

	PESO	VALOR
ITEM 1	7	10
ITEM 2	2	1
ITEM 3	5	6

A saída será a seguinte:

	SELECIONADO (0 ou 1)
ITEM 1	1
ITEM 2	1
ITEM 3	0

Portanto, o valor acumulado na mochila será de **13**.

6. PROBLEMA DA SUBSEQUÊNCIA COMUM MÁXIMA

Dadas duas sequências de caracteres A e B, a subsequência comum máxima (LCS, do inglês) é a maior série de caracteres na mesma ordem que ocorre tanto em A quanto em B, sendo que esses caracteres não precisam ser estritamente consecutivos. O Problema da Subsequência Comum Máxima consiste em encontrar essa LCS.

Para a resolução deste problema, utiliza-se da abordagem de Programação Dinâmica (já discutida anteriormente).

Complexidade: $O(2^{m \cdot n})$, (sendo **m**, **n** o tamanho das strings de entrada).

Considere, por exemplo, uma string A = “cimento” e uma string B = “cinto”. A subsequência comum máxima (LCS) a essas duas strings será “cinto”.