

Trabalho Prático 02 – AEDS 1

Professora: Thais R. M. Braga Silva

Valor: 10 pontos

Data de Entrega: 11/12/2024

Forma de Entrega: PVANet (.zip ou .tar.gz)

O objetivo deste trabalho prático é permitir a avaliação do impacto causado pelo desempenho dos algoritmos em sua execução real. Vimos em sala de aula que existem problemas e algoritmos de complexidade exponencial, chamados de intratáveis. Nesses casos, os programas, ao serem executados, podem demorar uma quantidade de tempo não razoável para encontrar uma solução, dependendo do tamanho da entrada. Vamos observar, portanto, como isso ocorre na prática.

Para tanto, cada grupo fará uma nova implementação para a operação E - Redistribuição de Rochas do Trabalho Prático 1 (TP1), a qual chamaremos de “Problema do Compartimento”. Esse, como veremos, é um problema intratável, pois sua solução exata somente é possível através do cálculo e avaliação de todas as possíveis saídas, o que chamamos de força bruta. Em seguida, essa implementação deverá ser executada para diferentes tamanhos de entrada N , e o tempo gasto para que o programa termine, em cada caso, deverá ser medido.

No TP1, a operação E apenas redistribuía as Rochas Minerais entre as Sondas para equilibrar o peso entre as mesmas. Agora, nesta versão chamada “Problema do Compartimento”, a ideia é preencher o compartimento de cada Sonda maximizando o seu valor final, isto é, colocando no compartimento de cada sonda a maior quantidade de valor agregado, de acordo com o valor de cada uma das Rochas Minerais disponíveis, sem ultrapassar sua capacidade.

Em linhas gerais, esse problema recebe como entrada X Sondas com valor C , que representa a capacidade do compartimento das mesmas (em termos do peso máximo suportado por ele) e um conjunto das N Rochas Minerais extraídas por estas Sondas, cada uma delas (i) representada por uma tupla contendo seu peso (p_i) e seu valor/importância (v_i). Vamos considerar que cada Rocha Mineral, individualmente, possui sempre peso inferior à capacidade do compartimento, isto é, $\forall p_i, p_i \leq C$. Assim, por exemplo, uma possível entrada seria $X = 1$, $C = 5$ e $N = \{(1,2),(3,1),(2,9),(1,5),(4,5),(2,3),(1,7),(1,3),(4,8),(2,2)\}$. O conjunto N indica que a Rocha Mineral 1 tem peso 1 e valor 2, a Rocha Mineral 2 tem peso 3 e valor 1, a Rocha Mineral 3 tem peso 2 e valor 9, e assim sucessivamente. O objetivo é encontrar para cada Sonda o subconjunto de N com maior valor agregado, que não ultrapassa a capacidade do compartimento. Em caso de empate, dentre os subconjuntos mais valiosos, aquele com maior número de itens é melhor.

Utilizando inicialmente as N Rochas Minerais da entrada, o programa deve encontrar o subconjunto de maior valor possível para a Sonda 1. Em seguida, removendo as Rochas Minerais já colocadas nesta primeira Sonda, ele deve usar as Rochas Minerais restantes e encontrar o segundo subconjunto de maior valor para a Sonda 2, e assim sucessivamente, até que todas as Rochas Minerais e/ou Sondas tenham sido utilizadas. Não há problema se sobraem Rochas Minerais que não puderam ser colocadas no compartimento das Sondas ou se Sondas ficarem sem Rochas Minerais adicionadas.

ATENÇÃO: A implementação ser feita para este trabalho é independente daquela feita para o TP1, ou seja, você deve iniciar novo projeto na sua IDE para este trabalho prático. Entretanto, é esperado que você utilize os TADs simples criados para o TP1, ou seja, TAD Rocha Mineral, TAD Compartimento, TAD Sonda e TAD Lista de Sondas.

Existem diversas possíveis implementações para o Problema do Compartimento. Entretanto, as mais utilizadas requerem recursos de programação que ainda não foram estudados por vocês. Dessa forma, adotaremos uma estratégia mais direta, visto que o objetivo principal do trabalho é a avaliação de desempenho, e não o desenvolvimento do algoritmo. Cada grupo deverá implementar um programa, em linguagem C, da seguinte forma:

- Procurar na Web um algoritmo de combinação simples X a X para um conjunto de N elementos (combinação = subconjunto sem repetição, a ordem não importa). Esse algoritmo deve, portanto, produzir todas as possíveis combinações de tamanho X dentre os N elementos. Você deverá utilizar a linguagem C. Portanto, se o algoritmo encontrado estiver em outra linguagem de programação, vocês deverão convertê-lo para linguagem C.

ATENÇÃO: se necessário, segue um link de referência sobre os conceitos de Análise Combinatória. Veja que, neste trabalho, se pede especificamente uma implementação de combinação. <https://descomplica.com.br/blog/arranjo-combinacao-e-permutacoes/>

- Seu algoritmo deve declarar internamente o valor 3 e 40 para o número de Sondas (X) e a capacidade do compartimento (C), respectivamente, e receber como entrada um conjunto de N Rochas Minerais, cada uma com seu peso e valor (ver arquivo de entrada de exemplo). Faça a leitura dos N valores a partir de um arquivo. Neste arquivo, o tamanho de N deve estar na primeira linha. Em seguida, haverá N linhas, cada uma delas com um par de inteiros separados por um espaço em branco. O par de inteiros da i -ésima linha representa o i -ésimo item do conjunto de rochas minerais. O primeiro inteiro será o peso e o segundo o valor do item.

- Você deverá usar seu algoritmo de combinação para criar combinações de tamanho 1 até N com os itens do conjunto de rochas minerais, isto é, você vai criar todas as combinações de tamanho 1, depois todas de tamanho 2, em seguida todas de tamanho 3, e assim sucessivamente, até que chegue à todas as combinações de tamanho N (no caso, apenas 1). Para cada combinação resultante do passo anterior, calcule seu valor agregado e seu peso total, somando os respectivos valores de cada elemento. Exclua as combinações que ultrapassem a capacidade C do compartimento. Para as demais, armazene aquela que possua o maior valor e, ao final, apresente-a como solução exata do problema.

- Execute o programa para os seguintes valores de N : 10, 15, 20, 25, 50. Ao executar o programa, utilize uma ferramenta para medição do tempo de execução, como o comando *time* do Unix.

Faça um relatório final contendo uma breve explicação do código implementado (indicando como funciona o algoritmo de combinação utilizado, de onde foi obtido, como o seu programa calcula as várias combinações de tamanhos 1 até N e como escolhe aquela cujo valor agregado é o maior, sem exceder a capacidade do compartimento). O relatório também deverá conter as configurações de hardware e software da máquina que foi utilizada e os resultados dos tempos de execução para os valores de N indicados acima. É obrigatório o uso do template de relatório, conforme foi feito para o TP1.

Responda também no relatório à seguinte pergunta: seria razoável executar o seu algoritmo para valores de $N \geq 100$? Justifique a resposta.

O código implementado e o arquivo do relatório final deverão ser entregues até a data limite através do PVANet. Lembrem-se de colocar os nomes e matrículas dos dois integrantes da dupla.