

Trabalho Prático 02 - AEDS 1

Professora: Thais R. M. Braga Silva

Valor: 10 pontos

Data de Entrega: 07/11/2019

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 dupla fará uma implementação para o “Problema da Mochila”. Esse 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 valores de entrada N , e o tempo gasto para que o programa termine, em cada caso, deverá ser medido por meio de comandos do sistema operacional.

Em linhas gerais, esse problema recebe como entrada um valor C , que representa a capacidade da mochila (em termos do peso máximo suportado por ela) e um conjunto de N itens, cada um deles (i) representado por uma tupla contendo seu peso (p_i) e seu valor/importância (v_i) . Vamos considerar que cada item, individualmente, possui sempre peso inferior à capacidade da mochila, isto é, C . Assim, por exemplo, uma possível entrada seria $C = 5$ 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 o item 1 tem peso 1 e valor 2, o item 2 tem peso 3 e valor 1, o item 3 tem peso 2 e valor 9, e assim sucessivamente. O objetivo é encontrar o subconjunto de N com maior valor agregado, que não ultrapassa a capacidade da mochila. Em caso de empate, dentre os subconjuntos mais valiosos, aquele com maior número de itens é melhor.

Existem diversas possíveis implementações para o Problema da Mochila. 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 dupla deverá implementar um programa, em linguagem C, para o problema da Mochila 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.
- Seu algoritmo deve declarar internamente o valor 40 para a capacidade da mochila, isto é $C = 40$, e receber como entrada um conjunto de N elementos, a serem armazenados em um vetor chamado Itens. Faça a leitura dos N valores a partir de um arquivo. Neste arquivo, o tamanho do vetor 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 vetor. 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 vetor, 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 da mochila. 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: 50, 80 e 100. Ao executar o programa, utilize uma ferramenta para medição do tempo de execução, como o comando *time* do Unix.
- Faça um relatório final contendo o código implementado, uma breve explicação do mesmo (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 da mochila). 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.
- Responda também no relatório à seguinte pergunta: seria razoável executar o seu algoritmo para valores de N maiores do que 200? Justifique a resposta.
- O arquivo do relatório final (pode ser um .doc, .pdf, .zip, .rar ou .tar.gz) deverá ser entregue até a data limite através do PVANet. Lembrem-se de colocar os nomes e matrículas dos dois integrantes da dupla.