

Trabalho Prático 1

Laboratório de Algoritmos e Estruturas de Dados I

Prof^a. Natália Batista

Nome: _____ Total: 15 pontos

Instruções

O trabalho prático é individual e deverá ser entregue até o dia 19/04/2018, por *email*. A data da entrega é fixa e, para cada dia de atraso, será descontado 20% do valor do trabalho.

Este trabalho deverá conter uma documentação conforme a especificação do Anexo I. A parte prática deverá ser implementada em linguagem C no ambiente Linux. Lembre-se de incluir as instruções de compilação dos programas e eventuais arquivos auxiliares, como “makefiles”.

Para a entrega dos arquivos (documentação e códigos), siga as instruções contidas no Anexo II.

O Problema da mochila

Imagine que você acaba de ganhar um concurso cujo prêmio é escolher n itens quaisquer em uma loja. O i -ésimo item tem valor v_i reais e pesa w_i quilos ($v_i, w_i \in \mathbb{R}$). Você poderá levar tudo o que quiser, desde que os itens escolhidos caibam em uma mochila que pode carregar até W quilos.

O objetivo é determinar o conjunto de itens que você deve pegar de forma que consiga carregá-los todos dentro da mochila e o valor total dos itens seja maximizado.

Este problema é conhecido como o problema da mochila e tem inúmeras aplicações práticas.

O trabalho prático consiste da implementação de soluções para o problema da mochila, testes e análise de resultados, conforme descrito nas questões a seguir.

1. Apresente dois algoritmos (em português estruturado ou similar) para solucionar o problema da mochila, o primeiro utilizando a técnica de tentativa e erro e o segundo utilizando uma estratégia gulosa.
2. Qual a ordem de complexidade de cada algoritmo proposto?
3. Os algoritmos levam sempre à solução ótima? (Discuta a questão separadamente para cada algoritmo).
4. Implemente cada um de seus algoritmos. Os formatos de entrada e saída são descritos a seguir.

Formato de entrada dos dados

Os n itens da loja serão fornecidos em um arquivo texto que contém na primeira linha W (capacidade da mochila), na segunda linha n (número de itens) e nas linhas seguintes os itens, cada um em uma linha (contendo o peso seguido do valor). Os valores devem ser números inteiros. Abaixo um exemplo de arquivo de entrada:

```
10
4
1 5
2 10
4 2
6 6
```

Arquivo texto
de entrada.

Formato de saída dos dados

O programa deverá produzir duas saídas:

- 1) um arquivo texto contendo, nesta ordem:
 - para cada item escolhido para a solução, o número do item, seu peso e seu valor (uma linha por item)
 - uma linha contendo o somatório dos pesos dos itens escolhidos
 - uma linha contendo o somatório dos valores dos itens.
- 2) a impressão na tela do tempo total de execução do programa.

5. Faça a medição do tempo de execução para cada algoritmo. Para cálculo do tempo é obrigatória a utilização das funções `getrusage` e `gettimeofday`. Deve-se também distinguir entre os tempos de computação e tempos de entrada e saída. Comente sobre os tempos de usuário e os tempos de sistema e sua relação com os tempos de relógio.

a) Execute cada algoritmo para valores de n crescentes (exemplo: $n=10, 50, 100, 200, 300, 400, 500$). Para cada valor de n , faça experimentos com valores de W também variados mas limitados a 50% da soma dos pesos de todos os itens, isto é,

$$W \leq 0.5 * \sum_{i=1}^n w_i.$$

Para cada entrada, discuta os resultados obtidos com cada algoritmo em termos do tempo de execução e precisão da resposta. Mostre gráficos que contraste estas medidas para os dois algoritmos. Os valores de v_i e w_i devem ser gerados aleatoriamente.

b) Informe o tamanho do maior problema para o qual você conseguiu obter a solução ótima. Comente o resultado, indicando o motivo da limitação e faça uma estimativa do tempo necessário no caso de termos uma entrada 10 vezes maior que a do maior problema que você resolveu.