

Exercício de Programação 3

Laboratório de Algoritmos e Estruturas de Dados I
Prof^a. Natália Batista

Nomes: _____ Total: 10 pontos

Instruções

Este exercício de programação será realizado nas aulas práticas de 28/04/23 e 05/05/23. A entrega poderá ser realizada até 11/05/23, pelo sistema run.codes. O exercício poderá ser realizado individualmente ou em duplas.

A solução deve ser implementada pelos(as) próprios(as) alunos(as) e não será permitido a utilização de trechos de códigos de outras pessoas ou retirados da internet.

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 ou mais mochilas, que podem 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(s) mochila(s) e o valor total dos itens seja maximizado.

Este problema é conhecido como o problema das múltiplas mochilas e tem inúmeras aplicações práticas.

O exercício consiste na implementação de um algoritmo para verificar se uma determinada solução apresentada na entrada é uma solução ótima, viável ou inviável para o problema das múltiplas mochilas com os parâmetros fornecidos (quantidade de mochilas, capacidade das mochilas, peso e valor dos itens).

Por exemplo, suponha uma única mochila com capacidade de 10 Kg e os seguintes objetos:

| Peso (Kg) | Valor (R\$) |
|-----------|-------------|
| 1 | 20 |
| 2 | 5 |
| 3 | 10 |
| 8 | 40 |
| 7 | 15 |
| 4 | 25 |

Este problema possui mais de uma solução ótima, que são equivalentes:

- Solução 1: valor total igual a R\$ 60.
Soma dos valores: R\$ 20 + R\$ 40 = R\$ 60
Soma dos pesos: 1 Kg + 8 Kg = 9 Kg
- Solução 2: valor total igual a R\$ 60.
Soma dos valores: R\$ 20 + R\$ 25 + R\$ 10 + R\$ 5 = R\$ 60
Soma dos pesos: 1 Kg + 4 Kg + 3 Kg + 2 Kg = 10 Kg

Há também outras soluções que são viáveis em termos de peso total, mas não são ótimas, por exemplo:

- Solução 3: valor total igual a R\$ 45.
Soma dos valores: R\$ 40 + R\$ 5 = R\$ 45
Soma dos pesos: 8 Kg + 2 Kg = 10 Kg

E há soluções inviáveis, cujo peso ultrapassa a capacidade total da mochila, por exemplo:

- Solução 4: valor total igual a R\$ 65.
Soma dos valores: R\$ 40 + R\$ 25 = R\$ 65
Soma dos pesos: 8 Kg + 4 Kg = 11 Kg

Formato de entrada dos dados

A entrada do programa é um arquivo texto cujo nome deverá ser solicitado, conforme modelo a seguir:

Digite o nome do arquivo:

```
printf("Digite o nome do arquivo: ");
```

O arquivo texto deverá conter as informações sobre as mochilas e os n itens da loja, nesta ordem:

- na primeira linha: q (quantidade de mochilas)
- na segunda linha: W_1 W_2 ... W_q (capacidade das mochilas, separadas por espaços em branco)
- na terceira linha: n (número de itens)
- nas n linhas seguintes: peso e valor dos itens, separados por um espaço em branco, cada um em uma linha
- nas n linhas seguintes: a solução a ser verificada (1 para item escolhido, 0 para item ausente na mochila)

Todos valores são números inteiros. Veja a seguir um exemplo de arquivo de entrada, que corresponde à Solução 1 citada anteriormente:

```
1
10
6
1 20
2 5
3 10
8 40
7 15
4 25
1
0
0
1
0
0
```

Formato de saída dos dados

O programa deverá produzir a impressão de uma mensagem na tela conforme o resultado da verificação, de acordo com o seguinte modelo (não utilize acentuação nem cedilha na saída para este exercício):

Solucao otima.

ou

Solucao viavel mas nao otima.

ou

Solucao inviavel.

Exemplo com duas mochilas

Arquivo de entrada:

```
2
65 85
6
40 110
60 150
30 70
40 80
20 30
5 5
0 1
1 0
0 0
0 1
0 0
1 0
```

Saída:

Solucao otima.

Observações importantes:

- O programa deverá solicitar ao usuário o nome do arquivo a ser testado. O programa deverá permitir testes com quaisquer arquivos no formato especificado (qualquer quantidade de mochilas, itens, pesos das mochilas, etc).
- O programa deverá ser bem organizado, indentado e conter comentários explicativos relevantes.
- Os testes do programa entregue serão realizados pelo sistema run.codes, conforme as instruções de utilização do sistema que podem ser consultadas pelo SIGAA.
- O programa deverá compilar sem erros ou avisos (*warnings*) com o compilador *gcc/g++*. Programas que não compilarem ou que apresentarem erros de execução (falhas de segmentação, etc) não serão corrigidos. Não serão aceitos arquivos enviados por e-mail e nem '*prints*' da execução do programa como prova de seu funcionamento.
- O programa deverá ser desenvolvido em ambiente Linux, seguindo todas as instruções da disciplina prática.
- Se houver indícios de plágio no código fonte do programa, a nota final da atividade será zero e serão aplicadas as penalidades previstas no Regime Disciplinar Discente.