

INSTITUTO SUPERIOR TÉCNICO

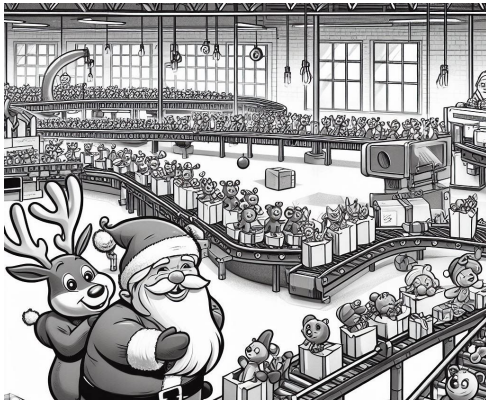
Análise e Síntese de Algoritmos

2023/2024

3º Projecto

Data enunciado: 18 de Dezembro de 2023
Data Limite de Entrega: 08 de Janeiro de 2024

Descrição do Problema



O Professor Natalino Caracol foi contratado pela empresa UbiquityInc, em Rovaniemi na Lapónia, para desenvolver um programa que permita estimar o lucro máximo que pode ser obtido com a produção e venda de brinquedos durante o Natal.

A empresa produz diariamente um conjunto de n brinquedos de madeira $\{x_1, \dots, x_n\}$, onde cada brinquedo x_i tem um lucro l_i . Para além de um limite máximo de produção de cada brinquedo devido a restrições de linha de montagem, a empresa está limitada a uma quantidade máxima total de brinquedos que podem ser produzidos por dia, devido a restrições de corte

da floresta boreal. Adicionalmente, este Natal a empresa decidiu, para além de vender cada brinquedo individualmente, vender também pacotes especiais contendo três brinquedos distintos, cujo lucro é maior do que a soma dos lucros individuais dos brinquedos que o constituem.

O objectivo consiste em indicar ao Rüdolf, CEO da UbiquityInc, qual o lucro máximo que se pode obter diariamente. A UbiquityInc, tratará posteriormente do problema da distribuição.

A implementação deve ser baseada em **Python com recurso à biblioteca PuLP** para resolução de problemas LP (<https://pypi.org/project/PuLP/>). Exemplos disponíveis em <https://github.com/coin-or/pulp/tree/master/examples>.

Nota: Informação sobre o que incluir no relatório pode ser encontrada no template a ser disponibilizado na página da disciplina.

Input

O ficheiro de entrada contém informação sobre os n produtos, o lucro e a capacidade de produção da empresa de cada um, da seguinte forma:

- Uma linha contendo três inteiros: t indicando o número de diferentes brinquedos passíveis de serem produzidos, p indicando o número de pacotes especiais, e max indicando o número máximo de brinquedos que podem ser produzidos por dia;
- Uma lista de n linhas, em que cada linha contém dois inteiros l_i e c_i , indicando o lucro e a capacidade de produção do brinquedo i .
- Uma lista de p linhas, em que cada linha contém quatro inteiros i , j , k , e l_{ijk} , indicando o lucro l_{ijk} do pacote especial $\{i, j, k\}$, e o nome dos produtos i , j , e k que o constituem.

Quaisquer inteiros numa linha estão separados por exactamente um espaço em branco, não contendo qualquer outro carácter, a não ser o fim de linha.

Output

O programa deverá escrever no output um inteiro correspondendo ao lucro máximo que o Rudolf pode obter diariamente.

Exemplo 1

Input

```
5 3 200
50 27
30 33
45 30
40 37
35 35
1 3 5 130
2 3 4 130
```

Output

```
6215
```

Exemplo 2

Input

```
5 3 200
50 27
30 33
45 30
40 37
35 35
1 3 5 130
2 3 4 130
```

Output

```
0
```

Implementação

A implementação do projecto deverá ser feita obrigatoriamente usando a linguagem de programação Python.

O tempo necessário para implementar este projecto é inferior a 15 horas.

A biblioteca PuLP deve ser instalada utilizando o comando:

```
python -m pip install pulp
```

Devem ter algum solver de PL instalado, como por exemplo o GLPK (<https://www.gnu.org/software/glpk/>) ou o LP_solve. Para instalar o GLPK (por exemplo no Ubuntu):

```
sudo apt-get install glpk-utils
```

Parâmetros de execução:

```
Python: python3 vossoprograma.py < ficheiro_de_input
```

Submissão do Projecto

A submissão do projecto deverá incluir um relatório resumido e um ficheiro com o código fonte da solução. Informação sobre as linguagens de programação possíveis está disponível no website do sistema Mooshak. A linguagem de programação é identificada pela extensão do ficheiro. Por exemplo, um projecto escrito em c deverá ter a extensão .c. Após a compilação, **o programa resultante deverá ler do standard input e escrever para o standard output.**

Informação sobre as opções e restrições de compilação podem ser obtidas através do botão help do sistema Mooshak. O comando de compilação não deverá produzir output, caso contrário será considerado um erro de compilação.

Relatório: deverá ser submetido através do sistema Fénix no formato PDF com não mais de 2 páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma descrição da solução, a análise teórica e a avaliação experimental dos resultados. O relatório deverá incluir qualquer referência que tenha sido utilizada na realização do projecto. Relatórios que não sejam entregues em formato PDF terão nota 0. Atempadamente será divulgado um template do relatório.

Código fonte: deve ser submetido através do sistema Mooshak e o relatório (em formato PDF) deverá ser submetido através do Fénix. O código fonte será avaliado automaticamente pelo sistema Mooshak (<http://acp.tecnico.ulisboa.pt/~mooshak/>). Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak e para o Fénix. Note que apenas a última submissão será considerada para efeitos de avaliação. Todas as submissões anteriores serão ignoradas: tal inclui o código fonte e o relatório.

Avaliação

O projecto deverá ser realizado em grupos de um ou dois alunos e será avaliado em duas fases. Na primeira fase, durante a submissão, cada implementação será executada num conjunto de testes, os quais representam 85% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 15% da nota final.

Avaliação Automática

A primeira fase do projecto é avaliada automaticamente com um conjunto de testes, os quais são executados num computador com o sistema operativo **GNU/Linux**. É essencial que o código fonte compile sem erros e respeite os standards de entrada e saída indicados anteriormente. Os projectos que não respeitem os formatos especificados serão penalizados e poderão ter nota 0, caso falhem todos os testes. Os testes **não serão divulgados antes da submissão**. No entanto, todos os testes serão disponibilizados após o deadline para submissão do projecto. Além de verificar a correcção do output produzido, o ambiente de avaliação **restringe a memória e o tempo de execução** disponíveis. A maior parte dos testes executa o comando `diff` da forma seguinte:

```
diff output result
```

O ficheiro `result` contém o output gerado pelo executável a partir do ficheiro `input`. O ficheiro `output` contém o output esperado. Um programa passa num teste e recebe o valor

correspondente, quando o comando `diff` não reporta quaisquer diferenças (i.e., não produz qualquer output). O sistema reporta um valor entre 0 e 170.

A nota obtida na classificação automática poderá sofrer eventuais cortes caso a análise do código demonstre recurso a soluções ajustadas a inputs concretos ou outputs aleatórios/constantes.

Detecção de Cópias

A avaliação dos projectos inclui um procedimento para detecção de cópias. A submissão de um projecto implica um compromisso de que o trabalho foi realizado exclusivamente pelos alunos. A violação deste compromisso ou a tentativa de submeter código que não foi desenvolvido pelo grupo implica a reprovação na unidade curricular, para todos os alunos envolvidos (incluindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, directa or indirecta, será comunicada ao Conselho Pedagógico do IST, ao coordenador de curso, e será penalizada de acordo com as regras aprovadas pela Universidade e publicadas em “Diário da República”.