

INSTITUTO SUPERIOR TÉCNICO

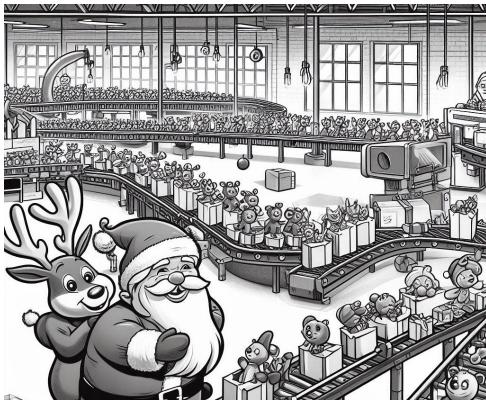
# Análise e Síntese de Algoritmos

2024/2025

## 3º Projecto

Data enunciado: 20 de Dezembro de 2024  
Data Limite de Entrega: 09 de Janeiro de 2025

### Descrição do Problema



O Professor Natalino Caracol foi contratado pela empresa do Pai Natal, a UbiquityInc em Rovaniemi na Lapónia, para desenvolver um programa que permita fazer a melhor distribuição possível de brinquedos de Natal pelas crianças de todo o mundo. A UbiquityInc segue uma estratégia descentralizada de produção de brinquedos por  $n$  fábricas,  $F = \{f_1, \dots, f_n\}$ , distribuídas distribuídas por  $m$  países,  $P = \{p_1, \dots, p_m\}$ , no mundo. Por razões de optimização da produção, cada fábrica  $f_i$  produz um único tipo de brinquedo e tem um stock máximo de brinquedos disponíveis para distribuição no Natal,  $fmax_i$ . Existem  $t$  crianças  $C = \{c_1, \dots, c_t\}$  espalhadas pelo mundo a enviar cartas ao Pai Natal, sendo que cada criança pede ao Pai Natal um conjunto de brinquedos, dos quais receberá, no máximo, um.

Por questões de equidade, a UbiquityInc estabeleceu para cada país  $p_j$  um número mínimo de presentes,  $pmin_j$ , a serem entregues nesse país. Adicionalmente, as regras de comércio internacional limitam as exportações do conjunto total das fábricas de cada país  $p_j$  a um valor máximo de  $pmax_j$ .

**Implementação:** O projecto deve ser baseado 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>.

**Relatório:** 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 toda a informação necessária sobre as  $n$  fábricas e os pedidos das  $t$  crianças, organizada da seguinte forma:

- Uma linha contendo três inteiros: o número total  $n$  de fábricas, o número total  $m$  de países, e o número total  $t$  de crianças;
- Uma lista de  $n$  linhas, em que cada linha contém três inteiros: o identificador  $i$  da fábrica  $f_i$  ( $1 \leq i \leq n$ ), o identificador  $j$  do país  $p_j$  ( $1 \leq j \leq m$ ) onde a fábrica se encontra, e um inteiro  $fmax_i$  indicando o stock da fábrica;
- Uma lista de  $m$  linhas, em que cada linha contém três inteiros: o identificador  $j$  do país  $p_j$  ( $1 \leq j \leq m$ ), um inteiro  $pmax_j$  indicando o limite de exportações de  $p_j$ , e um inteiro  $pmin_j$  indicando o número mínimo de brinquedos a distribuir em  $p_j$ ;
- Uma lista de  $t$  linhas, representando os pedidos de cada criança, em que cada linha contém três ou mais inteiros: o identificador  $k$  da criança  $c_k$  ( $1 \leq k \leq t$ ), o identificador  $j$  do país  $p_j$  onde  $c_k$  vive, e os identificadores das fábricas onde esses brinquedos são produzidos.

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

## Output

O programa deverá escrever no output um único inteiro correspondendo ao número máximo de crianças que poderão ver os seus pedidos satisfeitos respeitando as restrições do problema. Se não for possível satisfazer as restrições do problema deve ser impresso o valor  $-1$ . Em baixo, apresentamos alguns exemplos de inputs acompanhados dos outputs respectivos.

Input 1	Input 2	Input 3	Input 4
3 2 3	3 2 3	3 3 5	3 3 5
1 1 1	1 1 2	1 1 1	1 1 1
2 1 1	2 1 2	2 2 2	2 2 2
3 2 1	3 2 2	3 3 2	3 3 2
1 2 1	1 1 1	1 1 1	1 1 2
2 2 1	2 1 1	2 2 1	2 2 0
1 1 2 3	1 1 2 3	3 2 1	3 2 2
2 1 2 1	2 2 1 2	1 1 3	1 1 3
3 2 1	3 2 1 2	2 1 1	2 1 1
		3 2 3	3 2 3
		4 3 1 2	4 3 1 2
		5 3 1	5 3 1
Output 1	Output 2	Output 3	Output 4
3	2	4	-1

## 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”.