

INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos

2025/2026

3º Projecto

Data enunciado: 19 de Dezembro de 2025
Data Limite de Entrega: 09 de Janeiro de 2026

Descrição do Problema

O Eng. Norberto Caracol, cientista chefe da multinacional SnailSoft, é também um grande adepto de futebol. No entanto, tem-se sentido cada vez mais desiludido com a estratégia errática que tem observado a alguns treinadores de futebol, e com as previsões, por vezes pouco informadas, que tem ouvido de alguns comentadores desportivos. Assim, decidiu que a SnailSoft deverá desenvolver e comercializar uma aplicação informática que permita fornecer aos treinadores e comentadores desportivos informação analítica relevante sobre as competições, com uma base científica.

Esta aplicação destina-se a analisar competições (campeonatos) por pontos, que seguem as seguintes regras gerais:

- As competições terão obrigatoriamente duas voltas simétricas e as equipas participantes encontrar-se-ão todas entre si, uma vez na condição de visitados e outra na de visitantes, nos respetivos estádios.
- Em cada jogo os pontos serão atribuídos às equipas da seguinte forma: em caso de vitória, 3 pontos, em caso de empate, 1 ponto e, em caso de derrota, 0 pontos.

O objectivo é desenvolver um dos módulos desta aplicação informática que, dada a informação sobre as equipas participantes e sobre os resultados dos jogos já realizados, calcula, para cada equipa, qual o menor número de jogos que esta precisa de ganhar para ainda poder ganhar o campeonato, admitindo que os resultados dos jogos entre as outras equipas e os critérios de desempate são favoráveis.

Deverá modelar este problema com recurso a programação linear, e a sua implementação deverá 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>.

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 equipas participantes e os resultados dos jogos já realizados, organizada da seguinte forma:

- Uma linha contendo dois inteiros separados por um espaço: o número total, n , de equipas concorrentes e o número total, m , de jogos já realizados;
- Um conjunto de m linhas, em que cada linha descreve o resultado de um jogo já realizado, contendo três inteiros: o identificador i , $1 \leq i \leq n$, da equipa visitada (da casa), o identificador j , $1 \leq j \leq n \wedge j \neq i$, da equipa visitante e o resultado do jogo que será o identificador da equipa vencedora ou 0 caso o jogo tenha terminado num empate.

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 n linhas, em que cada linha de ordem i , $1 \leq i \leq n$, contém apenas um inteiro, que corresponde ao menor número de jogos que a equipa i precisa de ganhar para ainda poder ganhar o campeonato. Caso a equipa i já não possa ganhar o campeonato, deverá ser indicado o valor -1 .

Input 1	Input 2	Input 3
3 1	3 4	3 3
1 2 2	1 2 2	1 2 2
	2 1 2	1 3 3
	1 3 3	2 3 0
	3 1 3	
Output 1	Output 2	Output 3
1		2
0	-1	0
1	0	0
	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 (<https://codejudge.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.

Deteção 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”.