

INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

## 2023/2024

### 1º Projecto

Data enunciado: 14 de Novembro de 2023  
Data Limite de Entrega: 04 de Dezembro 2023

## Descrição do Problema

O Engenheiro João Caracol foi contratado pela fábrica SuperMarble para otimizar uma das suas linhas de corte de chapas de mármore. A linha recebe uma chapa de mármore que deve ser cortada de modo a produzir peças com dimensões requeridas pelos clientes da fábrica. A linha dispõe de uma máquina de dois discos que consegue cortar chapas de um lado ao outro. O processo de corte funciona da seguinte maneira:

- a chapa é cortada verticalmente ou horizontalmente;
- cada uma das duas novas chapas produzidas volta a entrar na linha de corte ou sai da linha, caso corresponda às dimensões de uma das peças a serem produzidas ou já não seja possível convertê-la numa peça.

A fábrica consegue neste momento escoar toda a produção pelo que deve ser dada prioridade ao fabrico de peças de valor mais elevado.

O objectivo do Eng. Caracol é construir um programa que dada uma chapa de mármore calcula o valor máximo que pode ser obtido a partir da mesma cortando-a em peças correspondentes às dimensões solicitadas pelos clientes. O Eng. Caracol pode produzir várias cópias de uma mesma peça conforme considere adequado. Mais concretamente: a linha recebe uma chapa rectangular de mármore de dimensões  $X \times Y$ . Além disso, o Eng. Caracol tem acesso a uma lista com os  $n$  tipos de peças a serem produzidos, todos com dimensões diferentes. Cada tipo de peça  $i \in \{1, \dots, n\}$  corresponde a um retângulo de mármore com dimensões  $a_i \times b_i$  e é vendido a um preço  $p_i$ .

## Input

O ficheiro de entrada contém as dimensões da chapa a ser cortada e as dimensões dos vários tipos de peças solicitados; estes valores são representados da seguinte forma:

- a primeira linha contém dois inteiros positivos  $X$  e  $Y$ , separados por um espaço em branco, que correspondem às dimensões da chapa;
- a segunda linha contém um inteiro positivo  $n$ , que corresponde ao número de tipos de peças que podem ser produzidas;
- $n$  linhas que descrevem cada um dos  $i$  tipos de peças que podem ser produzidas; cada linha é composta por três inteiros positivos  $a_i$ ,  $b_i$  e  $p_i$  separados por um espaço em branco, onde  $a_i \times b_i$  correspondem às dimensões do tipo de peça e  $p_i$  ao preço da mesma.

## Output

Deverá escrever no output o valor máximo que pode ser obtido a partir da chapa dada como input; caso nenhuma peça possa ser produzida deverá simplesmente imprimir 0.

## Exemplos

### Input 1

```
1 3
2
1 1 1
1 3 10
```

### Output 1

```
10
```

### Input 2

```
3 20
3
2 2 4
1 5 10
3 7 20
```

### Output 2

```
120
```

## Implementação

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C, C++ ou Rust. Submissões em linguagens interpretadas (Java/Python) também serão aceites, embora fortemente desaconselhadas. Alunos que o escolham fazer devem estar cientes de que submissões em Java/Python podem não passar todos os testes mesmo implementando o algoritmo correcto.

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

### Parâmetros de compilação:

```
C++: g++ -std=c++11 -O3 -Wall file.cpp -lm
C: gcc -O3 -ansi -Wall file.c -lm
Rust: rustc --edition=2021 -C opt-level=3 file.rs
Javac: javac File.java
Java: java -Xss32m -Xmx256m -classpath . File
Python: python3 file.py
```

## 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. Relatórios que não sejam entregues em formato PDF terão nota 0. 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. Atempadamente será divulgado um template do relatório como exemplo, contendo instruções adicionais sobre o seu conteúdo.

**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 expectedoutput result
```

O ficheiro `result` contém o output gerado pelo executável a partir do ficheiro `input`. O ficheiro `expectedoutput` 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).

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 deteção 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”.