

INSTITUTO SUPERIOR TÉCNICO  
Análise e Síntese de Algoritmos  
2017/2018

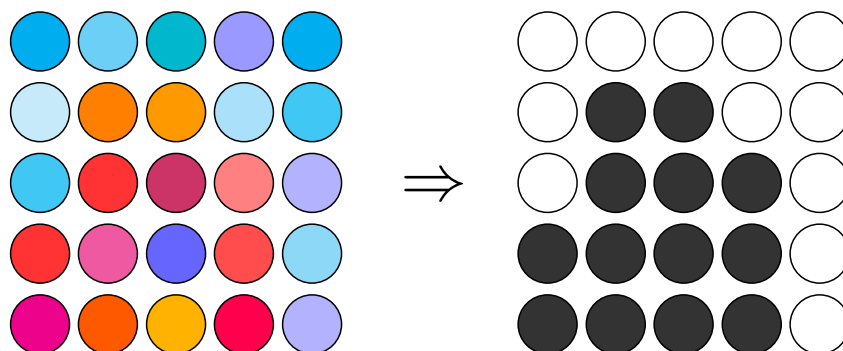
2º Projecto

Data Limite de Entrega: 02 de Maio de 2018

## Descrição do Problema

O Sr. Caracol é gerente de uma empresa de distribuição de mercadorias e anda bastante interessado nos recentes desenvolvimentos dos veículos sem motorista. Esta tecnologia poderia reduzir muito os custos de operação da sua empresa. Portanto o Sr. Caracol tem investido parte do orçamento da empresa na investigação de algoritmos relacionados com este tema. Um dos processos para os quais é necessário desenvolver algoritmos eficientes é a segmentação das imagens que são obtidas pela câmara. Vamos considerar uma versão simplificada deste problema.

Uma imagem é dada por um retângulo de píxeis. O objetivo é segmentar os píxeis, ou seja classificá-los como sendo de primeiro plano ou de cenário. As figuras abaixo ilustram um exemplo. A figura da esquerda mostra os píxeis com as respectivas cores. A figura da direita mostra a imagem segmentada, onde os píxeis brancos representam píxeis que fazem parte do cenário e os píxeis cinzentos representam píxeis de primeiro plano.



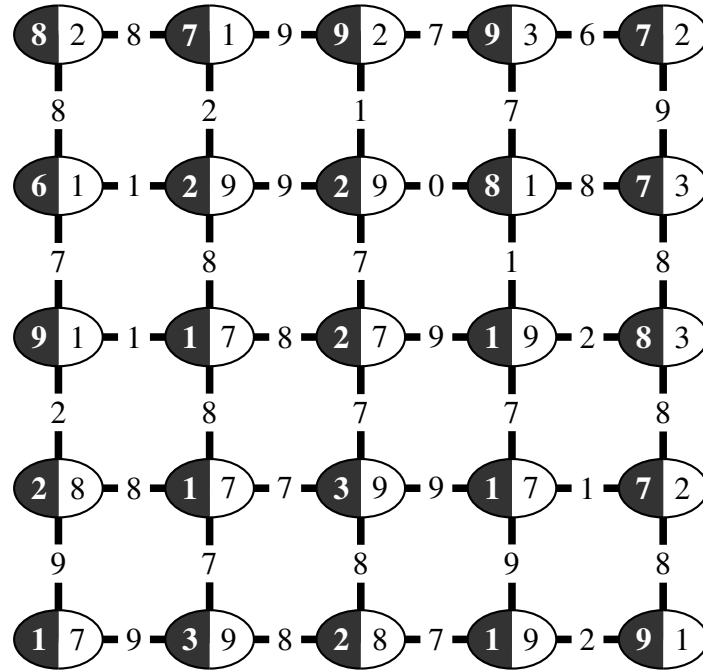
Existem vários algoritmos para executar este processo. Neste projecto iremos utilizar um modelo simples, que procura minimizar uma soma de peso. Estes pesos são determinantes para o resultado final e são obtidos por um processo estatístico.

Para uma imagem com dimensões  $m \times n$  cada píxel é identificado por um par  $(i, j)$  com  $0 \leq i < m$  e  $0 \leq j < n$ . O conjunto dos píxeis é representado pela letra  $P$ . Uma segmentação consiste na divisão dos píxeis no conjunto dos píxeis de primeiro plano  $L$  e no conjunto de píxeis de cenário  $C$ . Nenhum píxel deve ser simultaneamente de primeiro plano e de cenário, i.e.,  $L \cap C = \emptyset$ . Nenhum píxel deve ser omitido, i.e.,  $P = L \cup C$ . Cada píxel de primeiro plano,  $p \in L$ , contribui com um peso  $\ell_p$  para o peso total. Cada píxel de cenário,  $p \in C$ , contribui com um peso  $c_p$  para o peso total. Existem ainda pesos associados às relações de vizinhança. Cada píxel é vizinho do píxel acima, do píxel abaixo, do píxel à esquerda e do píxel à direita.

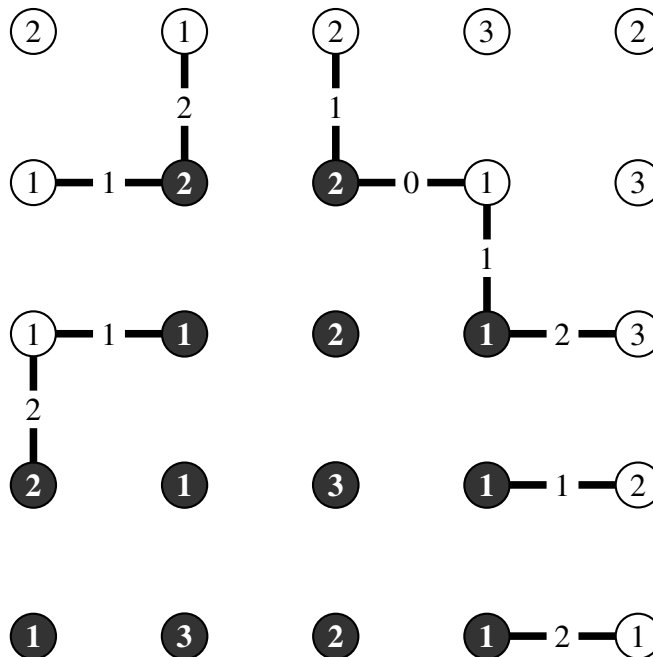
A cada relação de vizinhança  $v$  é associado um peso  $f_v$ . Quando existe uma relação de vizinhança  $v$  entre dois píxeis  $p$  e  $p'$  em que  $p \in L$  e  $p' \in C$  o custo  $f_v$  é adicionado ao peso total. A letra  $V$  representa o conjunto das relações de vizinhança com a propriedade anterior. O peso total de uma segmentação é calculada com a seguinte fórmula:

$$\left( \sum_{p \in L} \ell_p \right) + \left( \sum_{p \in C} c_p \right) + \left( \sum_{v \in V} f_v \right)$$

As figuras seguintes mostram um exemplo de uma segmentação. Os pesos de primeiro plano  $\ell_p$  estão nas zonas cinzentas dos píxeis, à esquerda. Os pesos de cenário  $c_p$  estão nas zonas brancas dos píxeis, à direita. Os pesos das relações de vizinhança estão sobre as respectivas relações.



A próxima figura ilustra uma segmentação dos píxeis. O peso total desta segmentação é  $22 + 22 + 13 = 57$ .



O seu programa deverá determinar a segmentação que minimiza o peso total. Será necessário reportar o peso total obtido e a respectiva classificação dos píxeis. Note que pode haver segmentações distintas que obtêm o mesmo peso total, nesse caso deve ser apresentada a segmentação que maximiza o número de píxeis classificados em primeiro plano. A segmentação apresentada nas figuras anteriores obtinha de facto o peso total mínimo e era única.

## Input

O ficheiro de entrada contém a informação do tamanho da imagem e dos pesos necessários.

O input é definido da seguinte forma:

- Uma linha com as dimensões  $m$  e  $n$  separadas por um espaço,  $m \geq 1$  e  $n \geq 1$ .
- Uma linha em branco, apenas com o caracter  $\backslash n$ .
- Uma sequência de  $m$  linhas com  $n$  inteiros cada, os inteiros são separados por um espaço e representam os pesos  $\ell_p$ .
- Uma linha em branco, apenas com o caracter  $\backslash n$ .

- Uma sequência de  $m$  linhas com  $n$  inteiros cada, os inteiros são separados por um espaço e representam os pesos  $c_p$ .
- Uma linha em branco, apenas com o caracter  $' \backslash n '$ .
- Uma sequência de  $m$  linhas com  $n - 1$  inteiros cada, os inteiros são separados por um espaço e representam os pesos associados às relações de vizinhança horizontais.
- Uma linha em branco, apenas com o caracter  $' \backslash n '$ .
- Uma sequência de  $m - 1$  linhas com  $n$  inteiros cada, os inteiros são separados por um espaço e representam os pesos associados às relações de vizinhança verticais.

## Output

O output é definido da seguinte forma:

- Uma linha com o peso total da segmentação mínima para a figura em questão.
- Uma linha em branco, apenas com o caracter  $' \backslash n '$ .
- Uma sequência de  $m$  linhas com  $n$  caracteres cada. Após cada caracter deve haver um espaço em branco, inclusivé após o último caracter. Os caracteres devem ser  $' P '$  ou  $' C '$ , conforme o píxel em questão é classificado como primeiro plano ou cenário, respectivamente.

## Exemplos

### input 1

```
5 5

8 7 9 9 7
6 2 2 8 7
9 1 2 1 8
2 1 3 1 7
1 3 2 1 9

2 1 2 3 2
1 9 9 1 3
1 7 7 9 3
8 7 9 7 2
7 9 8 9 1
```

8 9 7 6  
1 9 0 8  
1 8 9 2  
8 7 9 1  
9 8 7 2

8 2 1 7 9  
7 8 7 1 8  
2 8 7 7 8  
9 7 8 9 8

### output 1

57

C C C C C  
C P P C C  
C P P P C  
P P P P C  
P P P P C

### input 2

1 1

3

2

### output 2

2

C

### input 3

1 1

3

4

### **output 3**

3

P

### **input 4**

2 3

9 0 0

0 0 0

0 0 0

0 0 9

5 1

1 5

5 1 5

### **output 4**

3

C C P

C P P

### **input 5**

2 3

2 0 0

0 0 0

0 0 0

0 0 2

5 1

1 5

5 1 5

## output 5

2

P P P

P P P

## Implementação

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C ou C++. Submissões em linguagem Java também são aceitáveis, devendo no entanto ter particular atenção a aspectos de implementação.

O tempo estimado para implementar este projecto é inferior a 20 horas.

## Submissão do Projecto

A submissão do projecto deverá incluir um relatório resumido e um ficheiro com o código fonte da solução. Serão utilizadas as plataformas Mooshak para a submissão do código e Fénix para submissão do relatório. 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.

O relatório deverá ser entregue no formato PDF com não mais de 4 páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma introdução breve, a 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. O texto do relatório deve ser cuidado. Serão aplicados descontos por erros gramaticais ou erros ortográficos. Relatórios que não sejam entregues em formato PDF terão nota 0. O 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. Observe 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.

Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak e para o Fénix. Note que também é possível submeter várias vezes no Fénix e que não serão aceites relatórios fora de prazo e não haverá extensão de prazo.

O sistema Mooshak indica o tempo disponível para o projecto ser submetido. Os projectos têm que ser submetidos para o sistema Mooshak; não existe outra forma de submissão do projecto. Os relatórios têm que ser submetidos no sistema Fénix; não existe outra forma de submissão dos relatórios.

## 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 80% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 20% 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. Um conjunto reduzido de testes utilizados pelo sistema Mooshak serão públicos. A maior parte dos 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). Existem 16 testes. Assim, o sistema reporta um valor entre 0 e 16.

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, através do sistema moss (<https://theory.stanford.edu/~aiken/moss/>). 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 ou 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”.