**Análise e Síntese de Algoritmos**

**Projeto 2**

Grupo 21

João Martinho, 86454

Miguel Valério, 86483

# **Introdução:**

O presente relatório procura apresentar uma solução ao segundo projeto proposto para a cadeira de Análise e Síntese de Algoritmos de 2º semestre do ano letivo de 2017/18.

O projeto baseia-se numa empresa de distribuição de mercadorias que tem investido na investigação de desenvolvimento de veículos sem motorista. Para tal é necessário desenvolver algoritmos eficientes para a segmentação das imagens captadas pela câmara desses veículos.

Cada imagem é dada por um retângulo de pixéis. O problema é então segmentar os pixéis, isto é, segmenta-los como sendo de 1º plano ou de cenário.

# **Descrição da solução:**

Na resolução do problema, optámos pela linguagem de programação C.

A representação de uma imagem captada assenta na transformação do Input, constituído pelas dimensões da imagem (nº de pixéis por coluna e por linha), pelos valores de cada pixel pertencer ao 1º plano ou ao cenário e ainda pelos valores das relações de vizinhança entre pixéis (cada pixel é vizinho do pixel acima, do pixel abaixo, do pixel à esquerda e do pixel à direita), num grafo não dirigido (por cada aresta que ligue dois vértices existe uma que os liga no sentido oposto), onde os vértices são os diversos pixéis e as arestas representam as relações de vizinhança entre eles. Este grafo servirá como rede de fluxos para tal acrescentou-se ainda dois vértices: um a que chamamos P, que representa o 1º plano e que é a origem da rede, e outro que é fictício a que chamamos C, isto é, o vértice em si não existe, mas existem ligações para ele, este vértice representa o cenário. Todos os vértices ligam a P com o valor de pertencerem ao 1º plano e a C com o valor de pertencerem ao cenário.

O grafo foi representado sob a forma de uma matriz de adjacências. O conteúdo desta matriz é o peso das várias ligações entre vértices. Visto que cada pixel só é vizinho dos pixéis acima, abaixo, à esquerda e à direita, então cada linha da matriz só possui 6 elementos (1 para cada seu vizinho e ainda um para P e para C). No entanto a linha do vértice P tem um elemento para cada vértice existente na rede de fluxo, com exceção de si mesmo e do vértice C (visto que P nunca liga a nenhum destes dois).

A segmentação dos pixéis foi obtida através da aplicação de uma variante do algoritmo de Edmonds-Karp.

O nosso programa realiza os seguintes passos:

1. Leitura do input e construção do grafo;
2. Aplicação do algoritmo adaptado de Edmonds-Karp, com a variante onde, por sabermos *a priori* que a maior parte dos vértices do grafo têm ligação da origem para o vértice e do vértice para o destino (caminhos mais curtos nas V primeiras BFS) fazemos logo o envio de fluxo da source para o target através destes caminhos, esgotando pelo menos uma das arestas source->vértice ou vértice->target. Aplicamos de seguida o Edmonds-Karp sobre o grafo resultante.

O corte é obtido quando na BFS não é possível obter um caminho para a target, sendo que os vértices descobertos pertencem ao lado da source e os restantes ao lado do target.

1. Cálculo do peso total da segmentação. Uma vez que do lado da source se encontram os vértices para os quais existe maior somatório de pesos de 1º plano, e do lado do target se encontram os vértices com maior somatório de pesos de cenário, os pesos considerados para o peso total são os pesos de cenário dos vértices do lado da source e os pesos de 1º plano dos vértices do lado do target. As ligacões entre lados são obtidas verificando quais vértices de 1º plano ligam aos de cenário, considerando os pesos desses arcos.
2. Escrita de output, na ordem peso total da segmentação e representação do grafo em forma de matriz, com os vértices representados por um “P” se forem de 1º plano ou com um “C” se forem de cenário

# **Análise Teórica**

A complexidade temporal do programa desenvolvido, por se basear no algoritmo de Edmonds-Karp, e uma vez que para cada vértice (exceto a source) existem no máximo 6 ligações, estimamos que será de complexidade O(V3).

🡨------------------------------------------WE’RE HERE---------------------------------------------------🡪

Quanto à complexidade espacial, como no máximo são guardados uma lista de adjacências para cada vértice, isto é, guarda-se espaço para V vértices, e espaço para E arestas, estima -se que a complexidade será O(V+E). São também guardados outros vetores de tamanho V mas o limite assintótico continua a ser O(V+E).

# **Análise Experimental**

Para a análise experimental foram corridos 5 testes (5 grafos diferentes dados como input). Os testes foram corridos numa máquina com processador Intel® Core™ i7-7700HQ CPU @ 2.80GHz 2.81GHz, através de uma máquina virtual com 8GB de memória RAM, com sistema operativo Ubuntu (distribuição Linux).

Os testes obtiveram os seguintes resultados experimentais:

|  |  |  |
| --- | --- | --- |
| **V+E** | **Tempo (s)** | **Espaço (B)** |
| 150 | 0.001 | 8 696 |
| 1 500 | 0.001 | 40 000 |
| 15 000 | 0.025 | 343 120 |
| 90 000 | 0.1821 | 2 047 968 |
| 150 000 | 0.3172 | 3 466 880 |

**Nota:** Os resultados obtidos são uma média de 10 experiências executadas para cada teste.

Com os resultados apresentados anteriormente foi possível desenhar os seguintes gráficos:

Tal como previsto na análise teórica, o espaço de memória reservado durante a execução dos testes cresce linearmente com o número de vértices e arestas dos grafos. Portanto, a complexidade espacial é O(V+E).

Tal como previsto na análise teórica, o tempo a execução dos testes cresce linearmente com o número de vértices e arestas dos grafos. Portanto, a complexidade temporal é O(V+E).

# **Referências:**

As referências consultadas para a realização deste projeto foram:

* **Introduction to Algorithms, Third Edition:**Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford SteinSeptember 2009 ISBN-10: 0-262-53305-7; ISBN-13: 978-0-262-53305-8
* <https://pt.wikipedia.org/wiki/Pesquisa_binária>