

---

# Análise e Síntese de Algoritmos 2016/2017

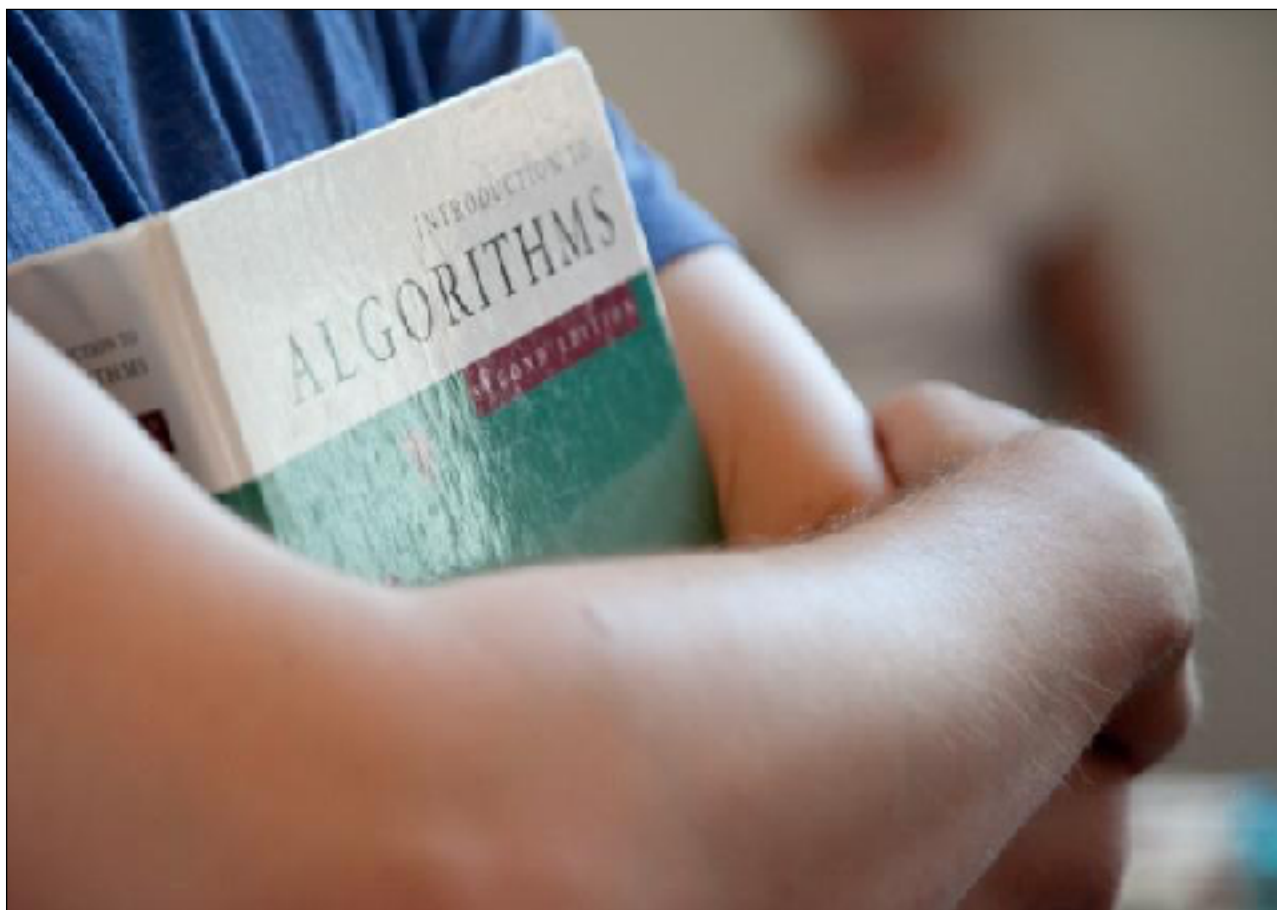
## 2º Projecto - Relatório

---

Christian Sousa - 67381

João Mak - 69613

Grupo 96



---

## Introdução

Este relatório discute a implementação do segundo projecto de Análise e Síntese de Algoritmos. O problema proposto envolve a verificação da existência de uma solução para a construção de uma rede de transportes (que pode ser feito por estrada ou por aeroporto) entre cidades, tendo como input a informação correspondente às estradas e aeroportos que podem ser construídos e respectivos custos. Para além disso, se for verificado que é possível solucionar o problema, deve ser devolvida a solução que implique o melhor preço (menor custo de construção de estradas e aeroportos), utilizando para isso o algoritmo Kruskal [1].

## Descrição da Solução

A implementação foi programada em Standart C++. Foram criadas as estruturas Graph e Edge para a implementação do problema.

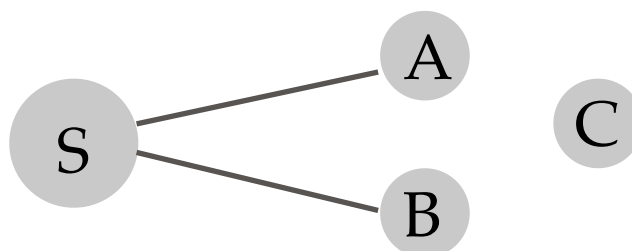
A estrutura Edge contém quatro inteiros **u**, **v**, **w** e **z**:

- **u** - corresponde ao identificador de uma cidade;
- **v** - corresponde ao identificador de uma cidade;
- **w** - corresponde ao custo de construção de uma estrada/aeroporto;
- **z** - identifica se o custo de construção corresponde a um aeroporto(**z=1**) ou uma estrada(**z=0**).

**Main()** - Aqui é feito o tratamento do input, criando dois grafos **g** e **e** :

- No grafo **e** os vértices do grafo correspondem às cidades e os arcos correspondem à ligação entre as cidades, contendo nesses arcos o valor do custo de construção de uma estrada que ligue duas cidades. Neste grafo apenas são consideradas as ligações entre cidades feitas por estradas.

- No grafo **g** são consideradas as ligações entre cidades, feitas tanto por estradas como por aeroportos. Para tal, e recorrendo a um vértice extra que representa uma localização fictícia **S**, conseguimos ter desta forma uma maneira de representar as ligações entre cidades utilizando aeroportos (isto porque qualquer cidade que tenha um aeroporto pode ser ligada a outra que também tenha aeroporto). Deste modo teremos como vértices os identificadores de cidade e como arcos as ligações entre elas assim como o custo de construção de estradas/aeroportos. No exemplo abaixo, as cidades **A** e **B** podem ser ligadas através de aeroportos se e só se estiverem ambas ligadas ao vértice **S** (**C** não estaria ligada a nenhuma das cidades por aeroporto).



---

Após o tratamento do input e da organização do mesmo, é corrido o algoritmo Krukal para os dois grafos.

Se, após terminada a execução dos dois algoritmos, não seja possível a criação de uma rede em que todas as cidades existentes fiquem ligadas, então a execução do programa termina e é devolvida a mensagem “Insuficiente”.

Caso contrário, e de forma a dar prioridade a redes onde o número de estradas a serem construídas seja superior ao número de aeroportos a serem construídos é comparado o valor do custo total da rede (MST - *Minimum Spanning Tree* - Árvore Abrangente de Menor Custo):

- Se o valor da MST correspondente apenas à rede de Estradas for menor ou igual à da MST do conjunto Aeroportos+Estradas, então é devolvido o valor do custo da MST da rede de estradas, bem como o número de Aeroportos e de Estradas a serem construídos;
- Se o valor da MST correspondente apenas à rede de Estradas for maior que a da MST do conjunto Aeroportos+Estradas, então é devolvido o valor da MST do conjunto Aeroportos+Estradas assim como o número de Aeroportos e Estradas a serem construídos.

## Análise Teórica

Começamos por tentar modelar o problema em termos algorítmicos, de forma a que a compreensão do problema e a formulação da melhor solução fosse facilitada. Decidimos que este problema podia ser exprimido como um problema de análise de um grafo, mais especificamente, a criação e análise de uma árvore abrangente de menor custo - MST.

Foi feito então o paralelismo:

- Cada nó/vértice corresponde a uma cidade;
- Cada arco (pesado e não dirigido) corresponde a um caminho entre as cidades, sendo o peso do arco interpretado como o custo associado à construção desse caminho (no caso de estradas/aeroportos, corresponde ao custo de construção de estrada/aeroporto);
- Uma MST correspondente a uma ligação entre todas as cidades com o menor custo possível.

Depois de estabelecido o paralelismo, tentámos perceber qual a solução óptima para o problema descrito. Assim:

- Para que houvesse uma facilidade de manipulação dos dados entre aeroportos e estradas (um input de aeroporto corresponde a um par cidade-custo e um input de estrada corresponde a um trio cidade-cidade-custo), criámos um vértice fictício S (correspondente a Sky ou céu), e assim uma cidade com aeroporto na verdade pode ser descrita como uma ligação cidade-céu-custo, o que também ajuda na compreensão do problema em si;

- Se não for possível criar uma ligação entre todas as cidades, o algoritmo teria de devolver “Insuficiente”;
- Se corrido correctamente um algoritmo MST (neste caso o algoritmo de Kruskal) então o resultado seria uma árvore abrangente de menor custo, em que o custo corresponderia ao menor custo monetário que permitisse que todas as cidades estivessem ligadas entre si;
- Como havia também a possibilidade de haver várias soluções óptimas e foi definido que se escolheria a solução que envolvesse menos aeroportos construídos, foi corrido o algoritmo Kruskal nos dois gratos **g** e **e**.

No problema desenvolvido, existem duas componentes com peso computacional considerável, a criação de grafos (  $O(E)$  ) e o Algoritmo de Kruskal (  $O(|E|\log|V|)$  ) concluindo-se portanto que a complexidade do algoritmo é de  $O(|E|\log|V|)$ .

## Avaliação Experimental dos Resultados

Para a verificação da eficiência da solução desenvolvida, recorreremos ao gerador de testes dados pelo corpo docente, assim como os exemplos que constam no enunciado do problema. Cada teste foi executado 5 vezes recorrendo ao comando de terminal “time ./gen\_p2 | ./a.out”. Realizou-se a média aritmética dos tempos obtidos, registando-se os seguintes resultados:

	10000 Cidades	100.000 Cidades	1.000.000 Cidades	Teste 3	Teste 4
<b>Tempo Real</b> (s)	0,047	0,347	4,618	0,015	0,027

## Referências

[1]. **Introduction to Algorithms, Third Edition:** Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein September 2009 ISBN-10: 0-262-53305-7; ISBN-13: 978-0-262-53305-8