

Relatório 2º projeto ASA 2022/2023

Grupo: AL057

Aluno(s): André Ferreira de Oliveira (103011) e Beatriz Mira Mendes (103120)

Descrição do Problema e da Solução

O problema proposto pretendia que através da análise a um grafo fosse possível calcular o máximo de trocas comerciais com o mínimo de custos. Para isso, começamos por construir o grafo obtido no input, guardando num vetor os arcos com as os seus respetivos vértices e pesos, e de seguida ordenando os arcos por ordem decrescente de peso.

De forma a calcular as trocas comerciais é utilizada uma implementação do algoritmo de Kruskal, onde a começando pelo arco mais pesado se vai testando se adição do mesmo criaria um ciclo infinito na MST. No caso de não criar, o peso do arco é adicionado ao peso da MST. Caso contrário o arco é ignorado e o peso da MST é mantido igual.

Análise Teórica

Numa fase inicial é lido o input de vértices (v) e de arcos (e) e são construídos o vetor de arcos e o mapa de vértices, através de dois loops, respetivamente $\Theta(e)$ e $\Theta(v)$. Assim pode-se concluir que a complexidade total na fase de leitura de inputs é de $\Theta(n)$. Pseudocódigo:

```
V = input;
E = input;
for E do
    leitura de inputs;
    introdução em vetor;
end for;
For V do
    inicialização de valor em map;
end for;
```

De seguida, é chamada a função sort, de complexidade $\Theta(e \log(e))$ e é feito um loop $\Theta(e)$, onde é testado se cada arco ao ser inserido na MST criaria um loop infinito. Para isso, são feitas duas chamadas a uma função find de complexidade $\Theta(n)$ e são feitas somas que também se comportam de forma constante. Uma vez que a função find é chamada 2ª vezes é esta que domina a complexidade deste processo, logo a complexidade da fase de tratamento dos dados é $\Theta(e)$. Pseudocódigo:

```
Sort(edges);
For E do
    find();
    find();
```

Relatório 2º projeto ASA 2022/2023

Grupo: AL057

Aluno(s): André Ferreira de Oliveira (103011) e Beatriz Mira Mendes (103120)

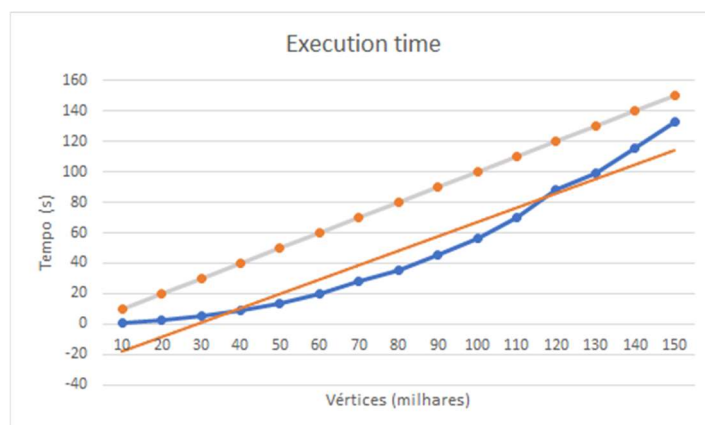
```
if smth do
    soma;
end if;
end for;
```

Por fim, o output consiste numa operação $\Theta(1)$, sendo possível assim concluir que a solução ao problema apresenta complexidade $\Theta(e) \Rightarrow \Theta(n)$.

Avaliação Experimental dos Resultados

Para testar a complexidade real da solução, utilizamos testes modelo para calcular o tempo de execução do programa quando o número de vértices aumenta. Utilizamos um incremento de mil vértices de um teste para outro. Os resultados foram os seguintes:

Vértices(centenas)	Tempo (s)
10	0,534
20	2,173
30	4,996
40	8,757
50	13,865
60	20,163
70	27,945
80	35,597
90	45,43
100	56,227
110	70,39
120	88,07
130	99,54
140	115,765
150	132,953



No gráfico é possível ver três linhas. A azul estão representados os resultados dos testes, a laranja a trendline que demonstra o comportamento do programa e a cinza uma linha que representa a função $f(x) = x$. Comparando a trendline e o comportamento da função $f(x)$ pode-se concluir que ambas têm um comportamento muito semelhante, o que comprova a previsão feita na avaliação teórica de que a solução apresenta complexidade $\Theta(n)$.