

Relatório 2º projecto ASA 2022/2023

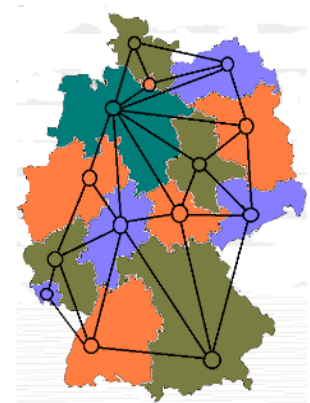
Grupo: AL037

Aluno: Diogo Rodrigues (102848)

Descrição do Problema e da Solução

Problema: Encontrar a forma de maximizar o número de trocas comerciais entre regiões vizinhas, usando o menor número possível de infraestruturas de transporte ferroviário, e garantindo que as trocas só ocorram entre regiões que estejam ligadas por uma rede ferroviária.

Solução: Encontrar uma MST (árvore geradora máxima) que representa a rede ferroviária que atenda aos critérios acima. Para isso, vamos utilizar o algoritmo de Kruskal, que funciona da seguinte forma:



1. Ordenamos as arestas (representando as trocas comerciais) de forma decrescente pelo seu peso (valor da troca).
2. Percorremos as arestas ordenadamente e, a cada passo, adicionamos a aresta selecionada à MST, desde que isso não crie um ciclo no novo grafo(MST).
3. No final, calculamos o peso total de todas as arestas da MST para obter o valor máximo de trocas comerciais.

Para suportar o algoritmo de Kruskal, usaremos árvores binárias como estrutura auxiliar.

Análise Teórica

Sendo “V” o número de vértices e “E” o número de arestas.

A leitura de dados consiste em ler o número de vértices e o número de arestas (tempo constante). De seguida lê-se “E” arestas (tempo linear). Logo, esta etapa tem uma complexidade $O(E)$, em tempo.

A aplicação do algoritmo de kruskal, pode dividir-se em 3 etapas:

- Operações make-set, $O(1)$, que ocorrem em tempo linear, $O(E)$.
- Ordenação das arestas que ocorre em tempo $O(E \log E)$, dependentemente do algoritmo de ordenação utilizado, no nosso caso, com a complexidade acima mencionada.
- Existem kE operações find-set, $O(V)$, e union, $O(1)$, no total. Logo esta etapa decorre em tempo $O(E\alpha(V))$.

Em suma, a complexidade do algoritmo de Kruskal corresponde à complexidade do algoritmo de ordenação utilizado, por ser a etapa com maior tempo de execução.

Relatório 2º projecto ASA 2022/2023

Grupo: AL037

Aluno: Diogo Rodrigues (102848)

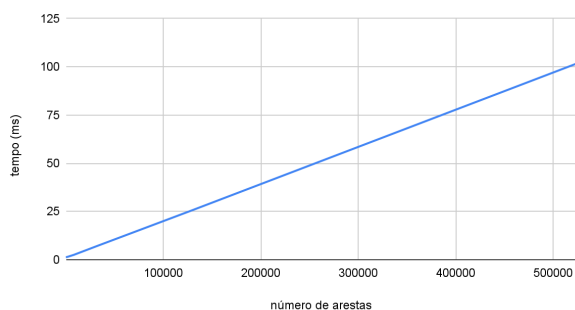
A apresentação de dados consiste em escrever o número de trocas comerciais máximo, que ocorre em tempo constante, $O(1)$.

Concluindo, a complexidade total deste programa, corresponde à complexidade da aplicação do algoritmo de Kruskal, devido a leitura e apresentação de dados terem complexidades inferiores. Assim sendo o programa tem um tempo de execução $O(E \log E)$.

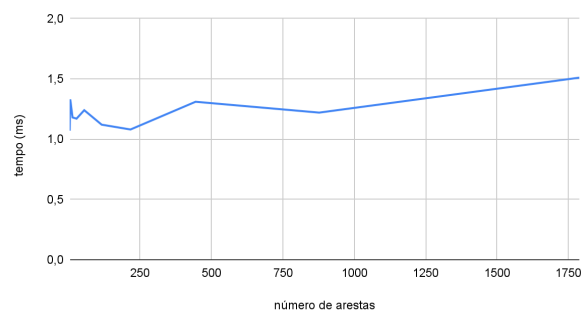
Avaliação Experimental dos Resultados

Foram feitos dois tipos de experiências para avaliar a complexidade do programa. Cada série consistiu em dez testes diferentes, e os gráficos mostram os resultados obtidos. O gráfico da esquerda apresenta os resultados dos testes com grafos densos, maior número de arestas possíveis, enquanto que o gráfico da direita mostra os resultados dos testes com grafos que respeitam a triangulação de Delaunay.

Grafo denso



Triangulação de Delaunay



Observando o gráfico da esquerda, inferimos que a complexidade do nosso algoritmo é linear, ao contrário do esperado, $O(E \log E)$. Uma possível explicação para isto é o facto do algoritmo de ordenação utilizado no caso geral, ser praticamente linear, para além do facto das arestas não estarem completamente desordenadas.

Quanto ao gráfico da direita, apesar do uso da ferramenta hiperfine, os resultados foram inconstantes pelo que não se pode tirar nenhuma conclusão.

*os testes gerados para a criação destes gráficos foram criados com as ferramentas disponibilizadas no fênix.