Relatório do 2º Projeto ASA 2024/2025

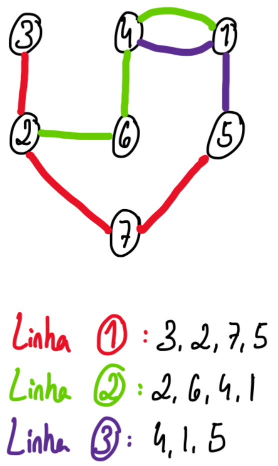
Grupo: AL019
Alunos: João Matreno (110246) e Samuel Gomes (110274)

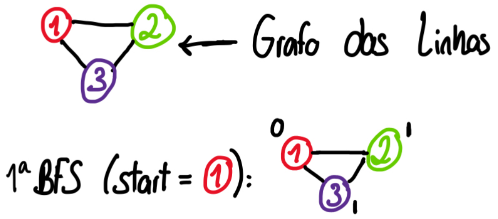
Explicação da solução

- **Construir a matriz de adjacência das linhas**, tal que, se duas linhas partilharem uma estação existe um arco entre elas no grafo, para isso mantemos umas lista das linhas a que um ponto pertence;
- Transformar a matriz numa lista de adjacência;
- Fazer uma **BFS nesse grafo**, e, sempre que se descobre uma nova linha, marcar as estações dessa linha como visitadas e colocar a sua distância como a distância da linha a que pertencem, isto se a estação ainda não tiver sido visitada. **Retorna a maior distância das estações**, ou "-1" caso alguma não tenha sido visitada;
- **Repetir** começando sempre num vertice diferente, desde que a linha correspondente não se encontre vazia de estações;
- Retornar o maior valor de todas as BFS's, ou caso alguma BFS dê "-1" ou "0", retornar esse valor imediatamente.

Exemplo:

```
7 8 3
3 2 1
2 7 1
7 5 1
2 6 2
6 4 2
4 1 2
4 1 3
1 5 3
```





Estações	1	2	3	4	5	6	7
Linha	2	1	1	2	1	2	1
Distância	1	0	0	1	0	1	0

Máx: 1

Análise Teórica da Solução Proposta

- Leitura dos dados de entrada e construção da matriz de adjacência: $\mathbf{O(ml)}$
  - número de ligações lidas: $\underline{m}$
  - para cada ligação existe a possibilidade de alguma das estações já pertencer a outra linha então conectamos todas as linhas dessa estação à nova linha, que no pior caso custa: $\underline{O(l)}$
- Transformação de matriz para lista de adjacências: $\mathbf{O(l^2)}$
- Aplicação do algoritmo indicado para cálculo do valor pedido: $\mathbf{O(ml + l^3)}$
  - número de vezes que chamamos a BFS: $\underline{l}$
  - a parte de atravessar o grafo das linhas tem um custo $O(l + l^2) \in \underline{O(l^2)}$
  - para cada linha com $x_l + 1$ estações são necessárias x_l ligações, então para visitar todas as estações temos $\sum_{l \in L}(x_l + 1) = m + l \in \underline{O(m)}$
  - vermos se todas as estções foram visitadas, e qual a maior distância registada: $\underline{O(n)}$
- Complexidade global da solução: $\mathbf{O(ml + l^3)}$

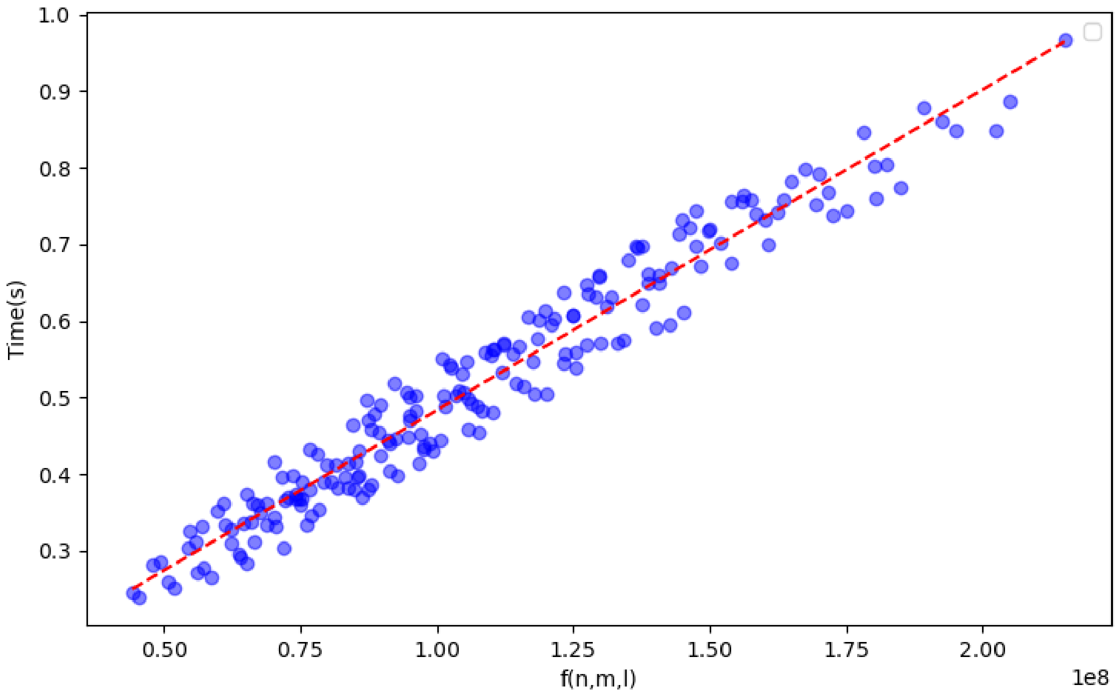
Relatório do 2º Projeto ASA 2024/2025

Grupo: AL019
Alunos: João Matreno (110246) e Samuel Gomes (110274)

Avaliação Experimental da Solução Proposta

Para fazer a análise experimental do nosso código medimos o tempo que demorava para encontrar a solução para inputs de diferentes tamanhos, tal que:

$$n = 150000 + 20000k_1,\ k_1 \in \mathbb{Z},\ 0 \leq k_1 \leq 9$$
$$m = 2n + 50000k_2,\ k_2 \in \mathbb{Z},\ 0 \leq k_2 \leq 5$$
$$l = 100 + 30k_3,\ k_3 \in \mathbb{Z},\ 0 \leq k_3 \leq 7$$



Neste gráfico temos que $\mathbf{f(n, m, l) = ml + l^3}$ e como podemos ver o tempo cresce linearmente em função de $f(n, m, l)$, o que confirma o esperado pela análise teórica de que a complexidade temporal do algoritmo é tal que $\mathbf{T(f(n, m, l)) \in O(ml + l^3)}$.

O que se verifica também é que a partir de um certo valor de l, com m constante, o tempo cresce linearmente com $\mathbf{l^3}$, mas, antes desse valor, o tempo cresce linearmente com $\mathbf{ml}$, confirmando mais uma vez o esperado pela análise teórica.