# Relatório 2º Projeto ASA 2024/2025

**Grupo:** AL063

**Alunos:** Madalena Mota (110355) e Ricardo Fonseca (109834)

---

## Descrição do Problema e da Solução

Na resolução deste projeto, começamos por armazenar as informações do input no formato inicial, o Grafo 1, que contém as linhas que passam por cada estação. Em seguida, processamos esses dados para construir o Grafo 2, em que os vértices são linhas e cada arco corresponde a uma ligação entre duas linhas.

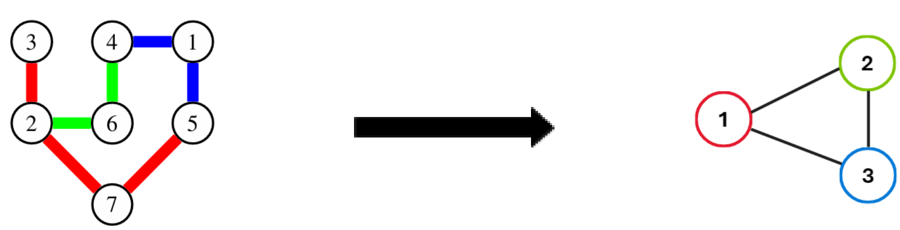
De seguida é aplicado o algoritmo BFS a cada vértice (linha) do Grafo 2 para obter a maior distância aos outros vértices. A distância representa, neste caso, o número de mudanças de linha efetuado. Deste modo, o resultado final do MC será o maior valor de distância de todos os retornos do BFS.

## Construção dos Grafos

<u>Construção do Grafo 1</u>: Para a primeira estrutura, utilizamos um vetor de sets (para não repetir elementos). Cada posição do vetor representa uma estação, por exemplo, a posição 1 representa a estação 1, a posição 2 representa a estação 2, etc. Para cada ligação que recebemos do input (*x*, *y*, *l*), adicionamos a linha *l* ao set de linhas de ambos os vértices da ligação, *x* e *y*. Armazenamos também todas as estações pertencentes a cada linha num vetor adicional (a sua utilidade será explicada de seguida).

<u>Construção do Grafo 2</u>: O Grafo 2 também consiste num vetor de sets. Desta vez, cada posição do vetor representa uma linha e contém as linhas que estão conectadas à mesma. Cada estação do Grafo 1 que contenha mais do que uma linha representa uma ligação entre todas essas linhas. Deste modo, para cada índice do Grafo 1 com mais do que um elemento, adicionamos todas as combinações entre esses elementos aos índices respectivos do Grafo 2.

Exemplo (Linha 1, Linha 2, Linha 3):



## Análise Teórica

- Construção do Grafo 1: O(mlog(nl)) - Recebemos um conjunto de **m** ligações e, para cada ligação, armazenamos a informação em 2 vetores de sets, onde:
    - A inserção de uma linha num set do vetor de estações possui complexidade O(log(l)), pois temos, no máximo, ***l*** linhas para cada estação.
    - Da mesma forma, armazenamos todas as estações pertencentes a cada linha num vetor e temos, no máximo, **n** estações para cada linha, pelo que a complexidade é O(log(n)).
- Edge Cases: Antes de construir o Grafo 2, fazemos duas verificações: se há uma estação desconectada, ou seja, não tem linhas, retornar -1 (O(n), pois percorremos o vetor de estações); se há uma linha que contém todas as estações, retornar 0 (O(l), pois percorremos o vetor de linhas).
- Construção do Grafo 2: O($nl^2$log(l)) - Temos **n** estações e, para cada estação, percorremos as linhas dessa estação (no máximo ***l*** linhas). Para cada linha, percorremos as linhas seguintes (no máximo ***l*-1** linhas). Para cada par de linhas inserimos um arco no grafo de linhas, que tem, no máximo, ***l*-1** linhas por set, pelo que a inserção é O(log(l)).
- Cálculo do MC: O($l^3$) - Utilizamos o algoritmo BFS para cada vértice no grafo das linhas. Deste modo, aplicamos o algoritmo ***l*** vezes. O algoritmo tem complexidade O($l^2$).

Complexidade global da solução: O(mlog(nl) + $nl^2$log(l)). $l^3$ é muito menor do que este valor, pelo que não influencia a complexidade.

## Avaliação Experimental dos Resultados

Para a avaliação experimental, foram testados 50 inputs distintos e crescentes. O gráfico resultante apresenta o tempo de execução em função da complexidade global obtida, f(n,m,l). Como esperado, estas grandezas têm uma relação linear, e o desempenho do algoritmo está em conformidade com a análise teórica.