Relatório 2º projeto ASA 2024/2025

Grupo: AL059

Aluno(s): Filipe Oliveira (110633) e Francisco Andrade (110720)

Descrição da Solução

São utilizados dois grafos: **grafo de estações e linhas** e **grafo de cores**.

- **Grafo de estações e linhas**: Os vértices são as estações e as arestas são as linhas que conectam as estações. Representado pela estrutura *MetroGraph* que contém duas subestruturas: *stationToLines* (mapeia cada estação para as linhas que passam por ela) e *lineToStation* (mapeia cada linha para as estações que ela conecta).
- **Grafo de cores**: Os vértices são as linhas de metro e as arestas são conexões entre linhas que partilham pelo menos uma estação. Representado por unordered_map e unordered_set (a chave é uma linha e o valor é o conjunto de linhas conectadas a ela).

Os dois grafos são construídos a partir do **input** da seguinte forma:

- **Grafo de estações e linhas**: Para cada ligação do input, é adicionada a linha ao conjunto de linhas que passam pelas duas estações (*stationToLines*) e as duas estações ao conjunto de estações conectadas pela linha (*lineToStation*). O resultado é o grafo *MetroGraph*.
- **Grafo de cores**: Construído a partir do MetroGraph através de uma função, conectando todas as linhas que partilham pelo menos uma estação. Para cada estação, são identificadas as linhas que passam por ela, e essas linhas são conectadas entre si no grafo de cores. Assim, se duas linhas compartilham uma estação, uma aresta é criada entre elas no grafo.

Como é calculado o **índice de conectividade** (**mc**)

A partir do *MetroGraph* são verificados os **casos específicos**: caso 0 (uma linha que cobre todas as estações) e caso -1 (existe uma estação desconectada ou o grafo de estações não é completamente conectado (usando BFS)).

Se não estamos perante um caso específico, procedemos ao **caso geral**, onde é construído o grafo de cores e calcula-se o maior caminho entre quaisquer duas linhas do grafo, usando BFS em cada linha. O resultado da aplicação da BFS é o **índice de conectividade**.

Análise Teórica da Solução Proposta

- **Leitura do input**: O programa lê três inteiros, *n* (estações), *m* (ligações), *l* (linhas), seguidos de uma lista com *m* entradas (ligações entre estações). Logo, **O(*m*)**.
- **Construção do grafo de estações e linhas**: Iteração pelas *m* ligações do input. É feita uma iteração por ligação, cada uma com O(1) para operações de inserção. Logo, **O(*m*)**.
- **Construção do grafo de cores**: Iteração sobre as *n* estações em *stationToLines*. Para cada estação são conectadas as k linhas que passam por ela. No pior caso, k pode ser proporcional ao número de linhas, *l*, resultando em $O(k^2)$ por estação. Logo, **$O(n * l^2)$**.
- **Cálculo do mc (casos específicos)**: No caso 0 é feita uma iteração sobre as *l* linhas e verificação do tamanho das estações conectadas, logo O(*l*). No caso -1 é feita uma iteração sobre as *l* linhas e inserção das estações conectadas num conjunto. No pior caso processa todas as *m* ligações, logo O(*m*). No caso -1 é verificada ainda a conectividade global, realizando uma BFS a partir de uma estação e processando cada

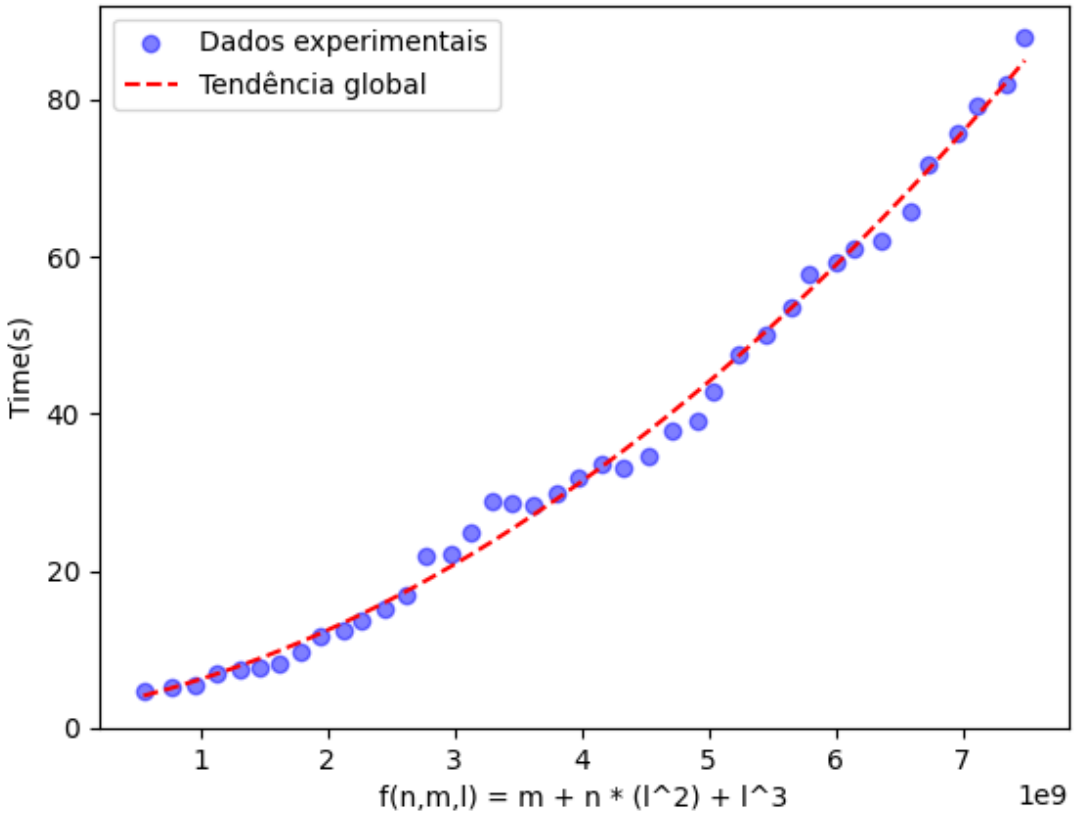
linha e estação uma vez, logo O(n + m). No pior caso $m \gg l$, logo a complexidade total é **O(m)**.

- **Cálculo do mc (caso geral)**: É realizada uma BFS a partir de cada linha no grafo de cores. Execução da BFS num grafo com l vértices e k arestas tem custo O (l + k). Como existem l vértices, no pior caso o custo é O (l * (l + k)). No pior caso, k pode ser O(l^2) (grafo completamente conectado). Logo, a complexidade total é **O(l^3)**.
- **Complexidade total**: Soma das etapas acima: O(m) + O(m) + O(n * l^2) + O(m) + O(l^3). No pior caso, a complexidade total é **O(m + (n * l^2) + l^3).**

Avaliação Experimental da Solução Proposta

De forma a fazer a avaliação experimental, foram testados 40 casos. Para obter o tempo de execução, foi utilizado o valor real do comando time. Foram utilizados os seguintes valores:

- ***n*** = 50k + 5k * (i + 1), o número de estações cresce linearmente, começando em 55k no primeiro teste e atingindo 255k no último;
- ***m*** = 100k + 10k * (i + 1), o número de ligações cresce linearmente, variando de 110k no primeiro teste até 510k no último;
- ***l*** = 100 + 20 * log (i + 1), o número de linhas apresenta um crescimento mais lento, variando de 100 a aproximadamente 170.



Com a representação do eixo dos XX de O(m + (n * l^2) + l^3), observamos uma relação linear entre a complexidade teórica prevista e os tempos registados. Logo, podemos concluir que o desempenho do algoritmo está em conformidade com a análise teórica, com uma complexidade assintótica de O(m + (n * l^2) + l^3).