Relatório 2º projecto ASA 2024/2025

Grupo: AL030

Aluno(s): Francisco Pestana (ist1109625)

Descrição da Solução

Explicação da solução: Para a resolução do problema foi utilizado um grafo de linhas, onde cada vértice corresponde a uma linha de metro diferente, e cada conexão representa o facto de existir transições entre estas linhas (ou seja, pelo menos uma estação em comum entre as linhas). Para isto é utilizada uma função auxiliar que retorna **true** caso duas linhas tenham estações em comum, e **false** caso contrário. O grafo é construído como um mapa **int->Vertice**, onde a chave é o valor inteiro identificador da linha, e o vértice é uma estrutura criada que armazena dados sobre o identificador da linha, as estações que pertencem à mesma, e as outras linhas as quais tem ligação.

Struct **Vertice** { int **line_id**; unordered_set<int> **stations**; unordered_set<int> **adjacencies**;}

Antes de o grafo ser construído, são verificados os casos limite (uma linha contém todas as estações, ou a rede de metro estar interrompida, ou uma estação fora da rede). <u>Depois</u> do grafo ser construído, o algoritmo **solve_MC** retorna a solução do problema, apresentando a seguinte logica:

for(vértice1 in grafo)
    for (vértice2 in grafo)
        if vertice1 == vertice2 continue
        if (commonStations(vertice1, vertice2) O(S)
            set_as_adjacent(vertice1, vertice2)
max_MC = BFS(grid)
return max_MC

Onde BFS realiza o algoritmo BFS para cada vértice do grafo, sempre mantendo o número máximo de mudanças necessárias entre cada iteração.

Análise Teórica da Solução Proposta

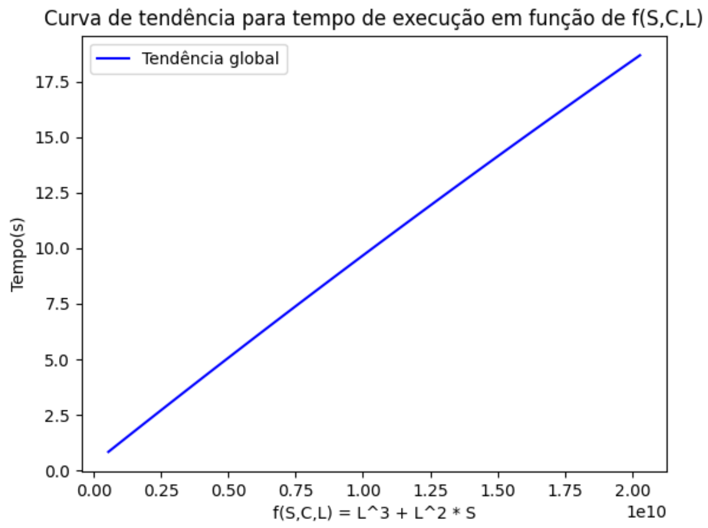
- **Leitura dos dados de entrada:**
  Ler S (número de estações), C (conexões) e L (linhas) O(1)
  Ler todas as conexões O(C)
  Verificação dos casos limite:
    - Ver se uma linha contem todas as estações O(L)
    - Ver se uma estação esta fora da rede O(C)
  Total: O(C + L)
- **Construção do grafo:**
  for (vertice1 in grafo)
    for (vertice2 in grafo) O(L^2)
      if vertice1 == vertice2 continue
      if (commonStations(vertice1, vertice2) O(S)
        set_as_adjacent(vertice1, vertice2)
  Total: O(L^2 . S)

- **Aplicação do algoritmo indicado para cálculo do valor pedido:** Dado que partimos de um algoritmo BFS de complexidade O(V + E), onde V é o número de vértices e E o número de arestas, no caso do problema, V corresponde a L (número de linhas) e E corresponde a L^2 (no pior caso, todas as linhas estão ligadas umas às outras). Isto resulta numa complexidade O(L^2). O algoritmo BFS é chamado para cada vértice do grafo, obtendo O(L^3 + L^2).
  Total: O(L^3)

- **Complexidade global da solução:** O($L^3 + L^2.S$) Resulta da complexidade da construção do grefo, mais a chamada ao algoritmo BFS.

Avaliação Experimental da Solução Proposta

Foram cronometrados sucessivos inputs de crescente números de estações, ligações e linhas. Estações de 5000 <= S <= 550000, com acréscimos de 5000; ligações de 100000 <= C <= 1100000 com acréscimos de 10000 e, linhas diferentes de 100 <= L <= 199 com acréscimos de 1.



Como previsto, a tendencia do tempo em função da complexidade estabelecida, apresenta uma natureza linear, o que demonstra a veracidade da complexidade prevista teoricamente