Relatório 2º projecto ASA 2024/2025

Grupo: AL046

Aluno(s): Guilherme Lopes (ist1109913) e Pedro Vicente (ist1109852)

Descrição da Solução (N - estações, M - conexões, L - linhas)

Grafo inicial

Construir um grafo inicial, “Graph”, que é um vetor com L unordered sets. Cada unordered set terá as estações que pertencem a essa linha. Se alguma das estações está isolada imprime -1 e se alguma linha contém todas as estações imprime 0.

Estrutura de dados usada para calcular o IC

Constrói-se um grafo reduzido, que será um vetor de L nodes, “Nodes”. Os nodes possuem atributos para criar o grafo: um inteiro que corresponde à linha que representam; um unordered set de inteiros, que correspondem às linhas a que estão conectados diretamente; um booleano, que indica se o node já foi criado no grafo. Possuem também atributos para serem usados numa BFS: um inteiro para distância; um booleano para representar se já foi visitado.

Como calcular a estrutura

Um loop for começa a iterar com “i” pelas L linhas de metro. Se não existir Nodes[i], criamo-lo. Um segundo for itera sobre as estações de Graph[i], uma “station” de cada vez. Um terceiro loop for itera com “j” pelo número de linhas de metro. Se não existir Nodes[j], criamo-lo. Verificamos se Graph[j] possui a “station”. Se possuir, as linhas estão diretamente ligadas; inserimos j em Nodes[i].adjacents e i em Nodes[j].adjacents. No final, verificamos se algum node tem o unordered set de adjacentes com tamanho 0. Se sim, significa que há uma linha isolada das outras e imprime -1.

Como calcular o IC a partir do grafo final

Com o grafo reduzido montado, e inicializando dois inteiros, connectivity e metro_connectivity, a 0, aplicamos BFS a cada node. A BFS altera o valor de connectivity para distância máxima entre o node de “start” e os outros nodes, seguindo o caminho mais curto possível entre estes. A variável metro_connectivity guarda o maior valor de connectivity e será impressa no fim do programa. Antes de retornar, a função da BFS verifica se algum node não é alcançável pelo node de “start”; se tal acontecer, a connectivity passa a -1 e é posteriormente impressa.

Análise Teórica da Solução Proposta

Indicar a complexidade de cada etapa da solução proposta, e a complexidade total. Por exemplo:

- **Leitura dos dados de entrada**: O(MN) – tirando a primeira linha, todas as outras representam uma das M conexões. Na construção do grafo original, são feitos inserts das estações no unordered set de cada linha que as contém. O insert nos unordered sets é O(N).

- **Construção do grafo**: O(N^2L^2) – o grafo reduzido consiste num loop for sobre as L linhas, outro loop for sobre as estações de cada linha, que é na pior das hipóteses N-1 (visto que, se uma linha tem N estações, imprimimos 0 e terminamos o programa), e um terceiro loop for, novamente sobre as L linhas. Ao verificar se Graph[j] tinha a “station” em causa, utilizamos o find, que é O(N). Logo, a complexidade de criar o grafo reduzido é O(N^2L^2).

- **Aplicação do algoritmo indicado para cálculo do valor pedido**: $O(M + NL^2 + L^3)$ ou O(M^3) –

  Casos 0 e -1

  Verificar se alguma linha tem todas as estações é O(1), pois ao ler o input verificamos se G[linha].size() == n. Ver se há estações isoladas também é O(1), já que verificamos se o size do unordered set “not_isolated_stations” é menor que N. Ver se há linhas isoladas é O(L), dado o loop for que contém a condição Nodes[linha].adjacents.size() == 0.

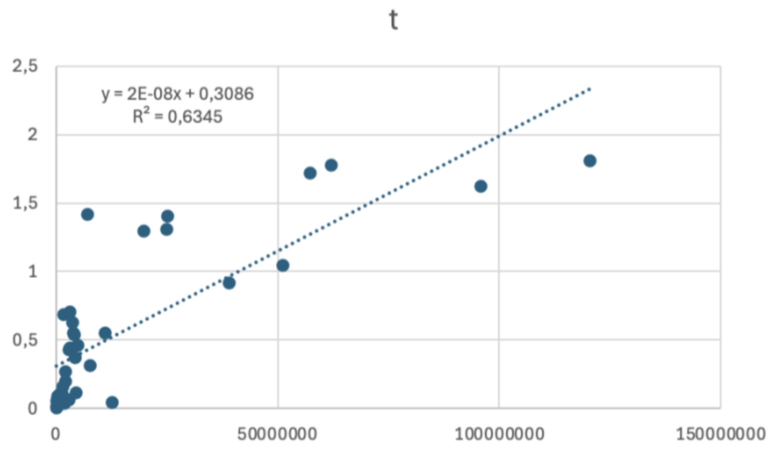
- Outros casos

  Nos outros casos, o número de arestas do grafo reduzido é, no pior caso, o de um grafo completo não dirigido, ou seja, $L(L-1)/2$. Assim, a análise agregada da BFS totaliza para O(V+E) = O($L + \frac{L^2 - L}{2}$) $= O\left(\frac{L^2+L}{2}\right) = O(L^2)$. Relembramos que a BFS é aplicada num ciclo for que percorre todos os L vértices do grafo reduzido, pelo que o ciclo totaliza para O(L^3).

Após analisados todos os passos para aplicação do algoritmo, vemos que este é O($MN + (NL)^2 + 1 + L + L^3) = O(MN + (NL)^2 + L^3)$. Isto vale para todo N, M, L, limitados pelos valores definidos pelo corpo docente.

- **Complexidade global da solução**: O($MN + (NL)^2 + L^3$) – Considerando os três parágrafos anteriores, temos $O(MN + (NL)^2 + L^3)$.

Avaliação Experimental da Solução Proposta



Como o grafo das linhas não é sempre um grafo completo, a complexidade máxima não é atingida em todos os casos. Este facto torna-se mais notável à medida que os valores de N, M e L aumentam.

Também é de salientar que, caso haja estações isoladas, ou inacessíveis a partir de outra estação, o programa termina mais cedo, não verificando assim a complexidade referida.