RELATÓRIO 2º PROJETO ASA 2023/2024

01 Descrição do Problema e da Solução

O problema abordado consiste em determinar o pior caso de propagação de uma dada infeção, calculando o número máximo de saltos que uma dada doença pode fazer, tendo em conta que indivíduos que se conhecem mutuamente, de forma direta ou indireta, ficam infetados automaticamente.

Na solução proposta modelámos as relações entre a população como um grafo dirigido. Assim, indivíduos que se conhecem mutuamente fazem parte do mesmo Componente Fortemente Ligado (SCC) e o número máximo de saltos da doença é o comprimento máximo do caminho entre os SCCs. Utilizamos uma abordagem baseada na busca em profundidade (DFS) para encontrar a ordenação topológica e, seguidamente, realizamos uma segunda DFS no grafo transposto de forma a identificar os SCCs. Finalmente, calculamos o comprimento máximo do caminho entre os Componentes Fortemente Ligados.

02 Análise Teórica

Seja n o número de indivíduos e m o número de relações a indicar.

- Leitura do input: primeiramente, ocorre a leitura do número de indivíduos e relações e, de seguida, sucede-se o ciclo para a leitura das relações entre indivíduos, que depende linearmente de m, pelo que a complexidade da leitura é $\Theta(m)$.

- Inicialização do vetor de SCCs: percorre todos os elementos uma vez, ou seja, $\Theta(n)$.

- Construção do grafo e do grafo transposto: cada aresta m é percorrida uma vez. Logo, $O(m)$.

- Inicialização do vetor *visited*: percorre todos os elementos uma vez, ou seja, $\Theta(n)$.

- Aplicação da primeira DFS iterativa: dentro de um loop que itera sobre todos os vértices n a função é chamada apenas para os vértices não visitados. Assim, cada vértice é visitado apenas uma vez. A função *dfsForward* tem um loop interno que executa enquanto há elementos na pilha. Cada aresta m do grafo é examinada apenas uma vez, pois, quando um nó é marcado como visitado, ele não volta a ser adicionado à pilha. Quando um vértice não tem vértices vizinhos por visitar, ele é removido da pilha e adicionado a uma outra pilha *st*, que regista a ordem topológica. Assim, a complexidade é $O(n + m)$.

```
function dfsForward(startNode, graph, visited, st)
  dfsStack = createEmptyStack()
  push(dfsStack, (startNode, 0))
  while not isEmpty(dfsStack)
    node, index = top(dfsStack)

    if index < size(graph[node])
      neighbor = graph[node][index]
      index = index + 1

      if not visited[neighbor]
        visited[neighbor] = true
        push(dfsStack, (neighbor, 0))
    else
      push(st, node)
      pop(dfsStack)
    end if
  end while
end procedure
```

- Reinicialização do vetor *visited*: $\Theta(n)$.

- Aplicação da segunda DFS iterativa: dentro de um loop que percorre a ordem topológica dos n vértices, a função é chamada se o vértice ainda não tiver sido visitado. Durante a DFS é atribuído o mesmo número de SCC a todos os vértices que são alcançáveis a partir do vértice inicial. O funcionamento geral é análogo ao da primeira DFS, ou seja, cada vértice n e aresta m são visitados apenas uma vez: $O(n + m)$.

- Cálculo do comprimento máximo do caminho entre os SCCs: a função começa por analisar as ligações entre os SCCs no grafo original, percorrendo cada aresta *m*. Seguidamente, calcula o comprimento máximo do caminho para cada SCC. Existem dois loops aninhados: o primeiro percorre os vértices e o segundo percorre os vizinhos de cada vértice. Assim, percorre todos os vértices *n* e arestas *m*. Por fim, encontra o índice do elemento máximo no vetor e retorna o valor associado a esse índice. A complexidade é $O(n + m)$.

- Apresentação dos dados: $\Theta(1)$.

Assim a complexidade global da solução é $O(n + m)$.

03 Avaliação Experimental dos Resultados

Para a avaliação experimental foram criadas 20 instâncias de tamanho incremental, sendo cada instância um grafo**,** em que *n* e *m* são incrementados 50 000 unidades a cada instância. Consequentemente $n + m$ registam um aumento de 100 000 unidades comparativamente à instância anterior.

Abaixo estão os resultados obtidos:

Como é possível concluir, existe uma relação linear entre o tempo e f(n, m), pelo que o resultado está de acordo com a previsão teórica.