Segundo trabalho de implementação – Análise e Projeto de Algoritmos

Primeiro parte: Dado um grafo formado por um conjunto de vértices e um conjunto de ligações desenvolver um algoritmo que verifique se o mesmo é conexo.

Caso não orientado:

Exemplo: Conjunto de vértices:

X = {x1.x2,x3,x4,x5,x6}

Conjunto de arestas:

U = {[x1,x2],[x2,x3],[x3,x1],[x4,x5],[x5,x6],[x6,x4]}

A implementação deve ser feita da seguinte forma:

1 – Construir a lista de adjacência a partir dos dados de entrada.

2 – Computar o fecho transitivo de um vértice escolhido, podendo ser qualquer um. Para tanto, utilizar o algoritmo de busca em profundidade. Não precisa visitar as arestas.

3 - Caso o conjunto do fecho seja igual a X o grafo é conexo, caso contrário desconexo.

Caso orientado:

Exemplo: Conjunto de vértices:

X = {x1.x2,x3,x4,x5,x6}

Conjunto de arcos:

U = {(x1,x2),(x2,x3),(x3,x1),(x4,x5),(x5,x6),(x2,x4)}

A implementação deve ser feita da seguinte forma:

1 – Simetrizar o grafo adicionando ao conjunto de arcos todos os arcos simétricos.

2 – Construir a lista de sucessores a partir dos dados de entrada.

3 – Computar o fecho transitivo direto de um vértice escolhido, podendo ser qualquer um. Para tanto, utilizar o algoritmo de busca em profundidade. Não precisa visitar os arcos.

4 - Caso o conjunto do fecho seja igual a X o grafo é conexo, caso contrário desconexo.

Segunda parte: Utilizar o prompt de uma LLM e pedir a geração do código de um ou dois algoritmos que resolva o problema anterior para os dois casos: orientado e não orientado. A entrada pode ser feita da mesma forma.