

## Implementação da Disciplina de Teoria dos Grafos

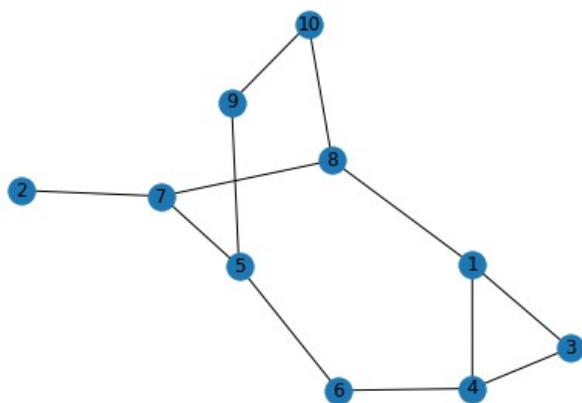
1- A proposta de implementação consiste na construção de uma aplicação para determinação das componentes biconexas de um grafo (ver Slides da Aula 7). O Algoritmo deve utilizar a busca em profundidade, tal como vista em sala de aula tem como passo intermediário a identificação das articulações. O algoritmo deve seguir fielmente os passos apresentados no slide da aula 7 e seção 4 livro do Jayme;

2- O Algoritmo deve construir o grafo com base em uma matriz de adjacência que será lida de um arquivo chamado “grafo.txt” com até 20 matrizes (ver exemplo). Cada matriz é separada por uma linha em branco;

```
0 0 1 1 0 0 0 1 0 0
0 0 0 0 0 0 1 0 0 0
1 0 0 1 0 0 0 0 0 0
1 0 1 0 0 1 0 0 0 0
0 0 0 0 0 1 1 0 1 0
0 0 0 1 1 0 0 0 0 0
0 1 0 0 1 0 0 1 0 0
1 0 0 0 0 0 1 0 0 1
0 0 0 0 1 0 0 0 0 1
0 0 0 0 0 0 0 1 1 0
```

2.1 – O arquivo “grafo.txt” pode conter várias matrizes de adjacências, de vários grafos, separadas por uma linha em branco. Dessa forma, logo depois de ler o arquivo **o programa deve informar quantas matrizes foram carregadas e qual delas o usuário deseja manipular** (a primeira matriz é a matriz 1, a segunda matriz é a matriz 2, a terceira matriz é a matriz 3, assim sucessivamente).

3 – Após ler a matriz de adjacência deve ser apresentado o grafo correspondente (ver exemplo abaixo);



4 – Na sequência deve-se apresentar o seguinte painel de seleção múltipla:

Digite a Opção Desejada:	
1	Apresentar Grafo (representação gráfica)
2	Apresentar Árvore de Busca em Profundidade
3	Apresentar Tabela Lowpt(v) e G(v)
4	Listar Articulações com seus Respective Demarcadores
5	Apresentar Componentes Biconexas e onde estão enraizadas (Tw)

- 5 – Caso o usuário selecione a opção “1” o programa deve apresentar o grafo com vértices numerados (na ordem em que aparecem da matriz de adjacência, com numeração iniciando em 1. Vértice  $v_1$  na linha 1;  $v_2$  na linha 2;  $v_3$  na linha 3; e assim sucessivamente);
- 6 – Caso o usuário selecione a opção “2” o programa deve perguntar: “Qual será o vértice raiz da busca?” e apresentar a listagem dos vértices candidatos; Na sequência deve-se ser apresentada a árvore de busca em profundidade com suas arestas de retorno (de preferência de cor diferente);
- 7 – Caso o usuário selecione a opção “3” o programa deve apresentar uma tabela com os mapeamentos de  $lowpt(v)$  e  $g(v)$ ;
- 8 - Caso o usuário selecione a opção “4” o programa deve apresentar a lista de todas articulações do grafo com os seus respectivos demarcadores;
- 9 - Caso o usuário selecione a opção “5” o programa deve apresentar cada componente biconexa do grafo (representação gráfica enraizada por  $T_w$ ).

\*A implementação deve ser original, i.e. não usar bibliotecas prontas para efetuar as principais tarefas solicitadas;

\*\* Todos algoritmos devem ser implementados tal como foram estudados em sala de aula;

\*\*\* Os conjuntos de arestas geradas pela busca em largura devem ser apresentados.