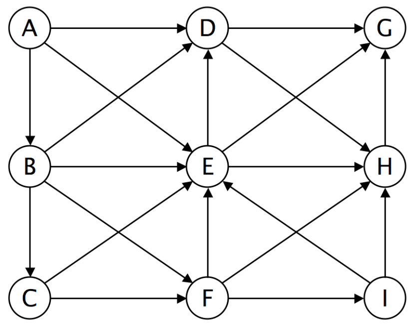
Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Informática (DAINF)
Professora: Juliana de Santi
Lista de exercícios (Busca em Grafos)

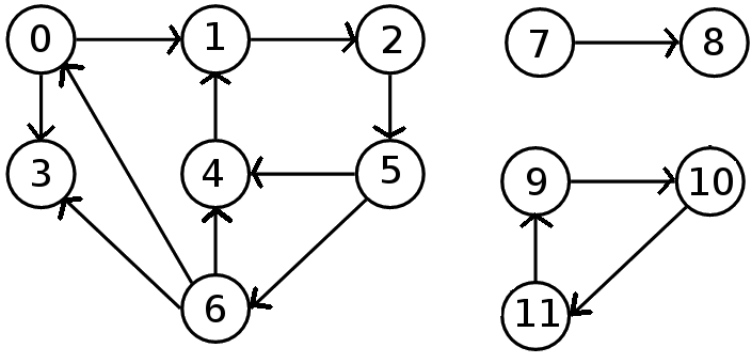
1)(Princeton University) Considere o seguinte grafo direcionado:



Considere a execução da busca comecando pelo vértice A. Assuma que as listas de adjacências estão em ordem lexográfica/alfabética, ou seja, ao explorar o vértice E, considere E-D antes de E-G ou E-H. Mostre: *a*)a lista de vértices da ordem de descoberta da busca em profundidade (DFS), e *b*) a lista de vértices na ordem em que eles são colocados na fila na busca em largura (BSF).

2) Implemente uma busca em profundidade em grafos (DFS) conforme visto em aula. Modifique o código da aula passada que utiliza uma lista de adjacência para codificar o grafo. Indique também, quais arestas do grafo são arestas de árvore e quais não são. Você deve após o término da DFS indicar todas as componentes encontradas pelo DFS (por exemplo, no grafo abaixo se a busca iniciar pelo vértice 0 ela irá visitar os nós 1, 2, 5, 4, 6 e 3). A definição de cores, predecessores, instante de descoberta e finalização deve ser feita fora da estrutura do Grafo (veja a estrutura DFS no código abaixo).

```
#define BRANCO 0
#define NIL -1
...
void Busca_Profundidade (GrafoA *G) {
   int u;
   DFS *V = (DFS *)malloc(G->V * sizeof(DFS));   <-  Estrutura independente!
   for (u = 0; u < G->V; u++) {
      V[u].cor = BRANCO;
      V[u].pai = NIL;
   }
   int tempo = 0;
   for (u = 0; u < G->V; u++) {
      if (V[u].cor == BRANCO) {
         DFS_Visit (G, u, V, &tempo);
      }
   }
}
```



Como exemplo, considere o grafo abaixo e a saída esperada:

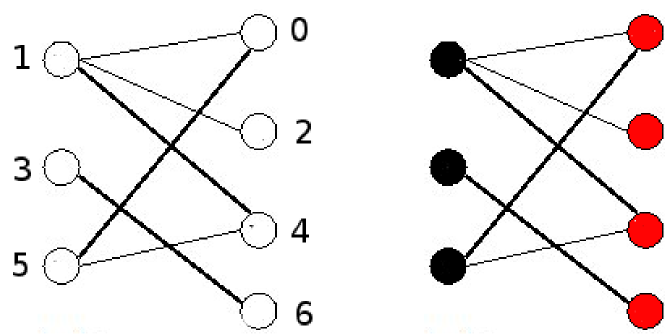
```
Saída esperada:

Aresta arvore: ( 0 -  1)
Aresta arvore: ( 1 -  2)
Aresta arvore: ( 2 -  5)
Aresta arvore: ( 5 -  4)
Aresta outra : ( 4 -  1)
Aresta arvore: ( 5 -  6)
Aresta outra : ( 6 -  0)
Aresta arvore: ( 6 -  3)
Aresta outra : ( 6 -  4)
Aresta outra : ( 0 -  3)
Aresta arvore: ( 7 -  8)
Aresta arvore: ( 9 - 10)
Aresta arvore: (10 - 11)
Aresta outra : (11 -  9)

Componente: 0   1   2   5   4   6   3
Componente: 7   8
Componente: 9  10  11
```

3) Em teoria dos grafos, um grafo bipartido ou bigrafo é um grafo cujos vértices podem ser divididos em dois conjuntos disjuntos U e V tais que toda aresta conecta um vértice em U a um vértice em V; ou seja, U e V são conjuntos independentes. Como analogia, suponha que seja necessário separar um conjunto de pessoas em dois grupos tal que dentro de um grupo não existam duas pessoas que se odeiam. O grafo abaixo é um exemplo de grafo bipartido:

Modifique a busca em profundidade (DFS) para que seja possível determinar se um dado grafo é bipartido ou não. Podemos modificar a DFS da seguinte forma: quando um novo vértice u for descoberto, atribua a ele uma cor oposta a de seu pai. Para toda aresta adjacente a u, verifique se u não tem algum vértice vizinho com a mesma cor, se ele tiver então o grafo é não bipartido. O primeiro vértice em qualquer componente conexa pode ser de qualquer cor.


1
3
5
0
2
4
6