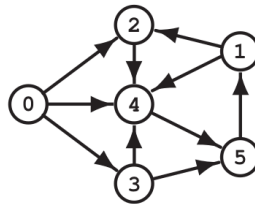


Lista de Exercícios – Grafos

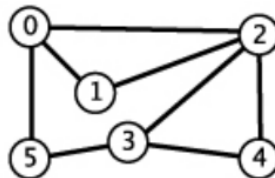
Busca em largura (breadth-first search) ou busca BFS

1) Considere o dígrafo G definido pelos arcos 0-2 0-3 0-4 1-2 1-4 2-4 3-4 3-5 4-5 5-1.

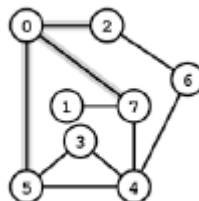


Suponha que os vértices estão em ordem crescente de nomes em cada lista de adjacência. Apresente a sequência de vértices visitados no percurso BFS.

2) Considere o Grafo não-dirigido definido pelas arestas 0-1 0-2 0-5 2-1 2-3 2-4 3-4 3-5. Suponha que os vértices estão em ordem crescente de nomes em todas as listas de adjacência. Apresente a sequência de vértices visitados no percurso BFS.



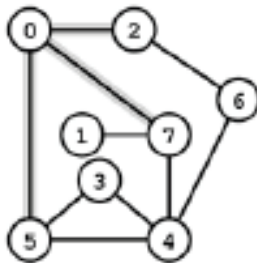
3) Faça uma busca em largura a partir do vértice 0 do grafo não-dirigido definido pelas arestas 0-2 2-6 6-4 4-5 5-0 0-7 7-1 7-4 3-4 3-5. Suponha que o grafo é representado por sua matriz de adjacências. Repita a busca começando pelo vértice 4.



Busca em profundidade (depth-first search) ou busca DFS

1) O grafo G tem conjunto de arestas 0-2 0-5 0-7 1-7 2-6 3-4 3-5 4-5 4-6 4-7 e portanto suas listas de adjacências:

0: 2 5 7
1: 7
2: 0 6
3: 4 5
4: 3 5 6 7
5: 0 3 4
6: 2 4
7: 0 1 4



Apresente a sequência de visitação dos vértices utilizando o algoritmo DFS.

2) Execute uma busca em profundidade no grafo dado pelas listas de adjacência a seguir.

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

Ordenação Topológica

1) Desenvolva um programa em linguagem C para implementar o algoritmo de ordenação topológica. Considere um dígrafo G com no máximo 30 vértices.

Material Complementar:

- 1) https://www.ime.usp.br/~pf/algoritmos_para_grafos/
- 2) Robert Sedgewick,
Algorithms in C
3rd. edition,
Addison-Wesley/Longman, 2002.

Resolver os exercícios do Urionline

1076 – Desenhando labirintos

1790 – Detectando pontes

1082 – Componentes conexos

1148 – Países em guerra

1774 – Roteadores

1152 – Estradas escuras

1128 – Ir e vir

2294 – Duende perdido

Resolver o exercício da OBI (anexo)

Pedágio