Assignment 2

CS-F222 Discrete Structure for Computer Science
Due Date: 23rd Nov, 2020

Total Marks: $10 \times 5 = 50$

- Implement programs for the following questions by assuming an input and to get required output. For every question place question, your code, followed by screenshots of two instances of input and the corresponding output, in a single pdf file.
- All your programs must be written in C. Assignments with code written in any other programming language than C will not considered for evaluation.
- You have to send your pdf file with your allotted serial number as the file name to your allotted TA (consult excel sheet uploaded on CMS for your allotted serial number and for your allotted TA) before the deadline.
- The TA's are S Vishwanath Reddy (p20190420@hyderabad.bits-pilani.ac.in) and B S A S RAJITA (p20150409@hyderabad.bits-pilani.ac.in)
- Your ID number and name are must on the first page of your document.
- Answer ALL Questions.

---

1. Let R be a relation on a set A. R may or may not have some property P, such as reflexivity, symmetry, or transitivity. If there is a relation S with property P containing R such that S is a subset of every relation with property P containing R, then S is called the closure of R with respect to P.
   Given the matrix representing a relation R on a finite set, find the matrix representing the reflexive closure, symmetric closure and transitive closure of R.

2. Given the matrix representing a relation R on a finite set, determine whether R is an equivalence relation or not.

3. Let R be a relation from a set A to a set B and S be a relation from B to a set C. The composite of R and S is the relation consisting of ordered pairs (a, c), where $a \in A$, $c \in C$, and for which there exists an element $b \in B$ such that $(a, b) \in R$ and $(b, c) \in S$. We denote the composite of R and S by $S \circ R$. The powers R^n , $n = 1, 2, 3, \ldots$, are defined recursively by $R^1 = R$ and $R^{n+1} = R^n \circ R$.
   Ex. Let $R = \{(1, 1), (2, 1), (3, 2), (4, 3)\}$
   $R^2 = R \circ R = \{(1, 1), (2, 1), (3, 1), (4, 2)\}$
   $R^3 = R^2 \circ R = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$ and so on.
   Given the matrix representing a relation R on a finite set and an integer n. Find the matrix representing R^n.

4. A total ordering $\preceq$ is said to be compatible with the partial ordering R if $a \preceq b$ whenever aRb. Constructing a compatible total ordering from a partial ordering is called topological sorting. The topological sorting algorithm works for any finite nonempty poset, to define a total ordering on the poset $(A, \preceq)$, first choose a minimal element a_1, next choose a minimal element a_2 of poset $(A - \{a_1\}, \preceq)$, then remove a_2 as well, and if there are additional elements left, choose a minimal element a_3 in $A - \{a_1, a_2\}$, Continue this process by choosing a_{k+1} to be a minimal element in $A - \{a_1, a_2, ..., a_k\}$, as long as elements remain. Given a partial ordering on a finite set, find a total ordering compatible with it using topological sorting.

5. A Cut vertex (Articulation point) of a graph is a vertex whose deletion along with incident edges results in a graph with more components than the original graph. A Graph G is said to be strongly connected if it does not contain any Cut vertex. Given an adjacency matrix of a graph G, check whether it is strongly connected or not. If not find all cut vertices of G.

6. A sequence $d_1, d_2, d_3, \ldots, d_n$ is called graphic if it is the degree sequence of a simple graph G (where d_i is the degree of v_i in the vertex-set of G. Given a degree sequence check whether it is graphical or not.

7. A coloring of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color. The chromatic number $\chi(G)$ of a graph G is the least number of colors needed for a coloring of this graph G Given an adjacency matrix of a graph, find its Chromatic number $\chi(G)$.

8. A simple path in a graph G that passes through every vertex exactly once is called a Hamilton path, and a simple circuit in a graph G that passes through every vertex exactly once is called a Hamilton circuit.

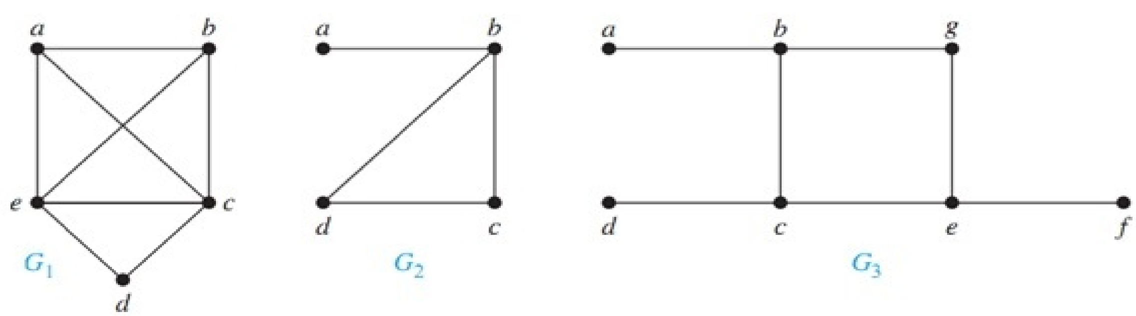

Figure 1

In Figure 1, G_1 has a Hamilton circuit: a, b, c, d, e, a. There is no Hamilton circuit in G_2 but G_2 does have a Hamilton path, namely, a, b, c, d. G_3 has neither a Hamilton circuit nor a Hamilton path.
Given the list of edges of a simple graph, produce a Hamilton circuit, or determine that the graph does not have such a circuit.

9. Depth-First Search (DFS): a procedure for constructing a spanning tree by adding edges that form a path until this is not possible, and then moving back up the path until a vertex is found where a new path can be formed. Example: see Figure 2.

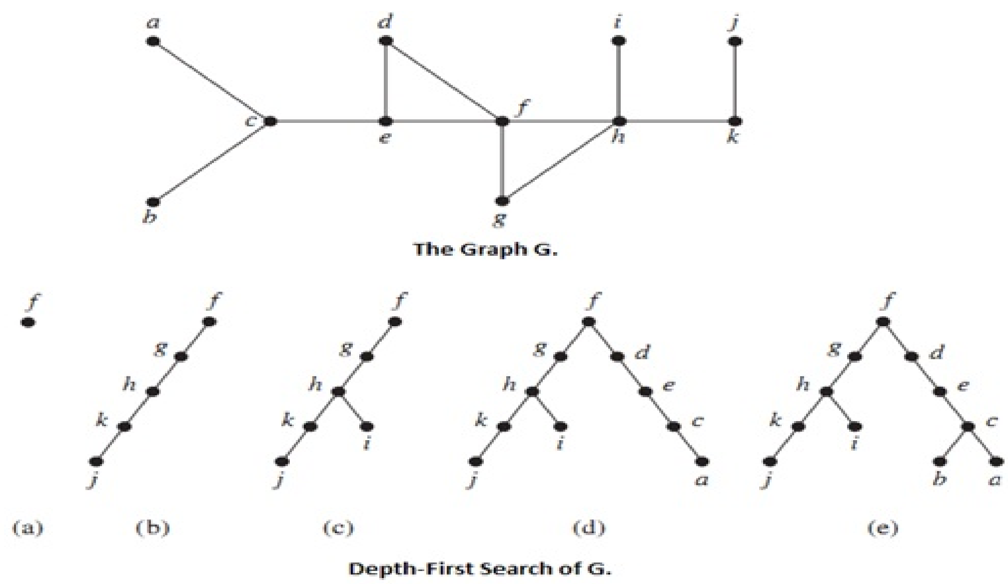

Figure 2

Breadth-First Search (BFS): a procedure for constructing a spanning tree that successively adds all edges incident to the last set of edges added, unless a simple circuit is formed.Example: see Figure 3.

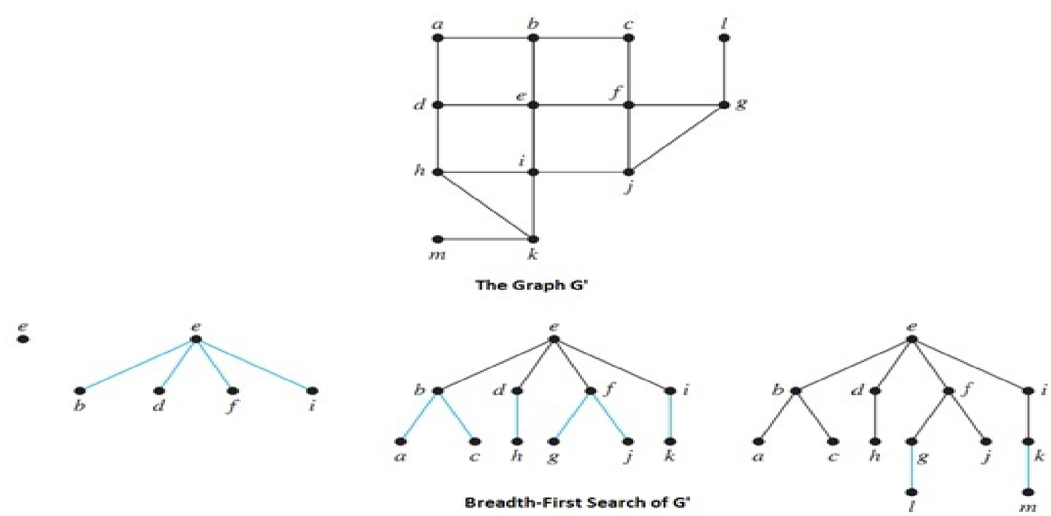

Figure 3

Given adjacency matrix representation of a connected undirected graph, produce depth first search (DFS) and breadth first search (BFS).

10. Given a positive integer n, list all the bit sequences of length n that do not have a pair of consecutive $0'$s.