

1. Show that the sum of degrees of all vertices in an undirected graph is twice the number of edges.
2. Write a program to Add and Remove Edge in Adjacency Matrix representation of a Graph.
3. How many edges are contained in a complete graph of “n” vertices?
4. Explain how existence of a cycle in an undirected graph may be detected by traversing the graph in a depth first manner.
5. Write a program in C to print the Minimum number of nodes in an AVL Tree with given height.
6. Give an example of a connected directed graph so that a depth first traversal of that graph yields a forest and not a spanning tree of the graph.
7. Rewrite the algorithms “BFSearch” and “DFSearch” so that it works on adjacency matrix representation of graphs.
8. Write a C function to find out whether there is a path between any two vertices in a graph (i.e. to compute the transitive closure matrix of a graph)
9. Write a C function to delete an existing edge from a graph represented by an adjacency list.
10. Solve exercise 8.3.1 from “Data Structures using C and C++” by Tennenbaum.
11. Solve exercise 8.3.2 from “Data Structures using C and C++” by Tennenbaum.
12. Solve exercise 8.3.3 from “Data Structures using C and C++” by Tennenbaum.
13. Solve exercise 8.3.4 from “Data Structures using C and C++” by Tennenbaum.
10. Solve exercise 8.4.1 “Data Structures using C and C++” by Tennenbaum.
11. Solve exercise 8.4.2 “Data Structures using C and C++” by Tennenbaum.
12. Solve exercise 8.4.7 “Data Structures using C and C++” by Tennenbaum.

- 
13. Solve exercise 8.4.8 “Data Structures using C and C++” by Tennenbaum.
  14. Write an algorithm and a C-Program to determine if a directed graph is a dag
  15. Write a recursive C-Program to print the nodes of the dag in reverse topological order
  16. Write a non-recursive C-Program to print the nodes of the dag in reverse topological order
  17. Write a c-routine that use a breadth first search traversal to determine if a directed and an undirected graph are cyclic.
  18. Implement a graph so that the lists of header nodes and arc nodes are circular.
  19. What is the difference between undirected and directed graph.
  20. What does it mean for a directed graph to be strongly connected? Weekly connected?
  21. Explain the difference between DFS and BFS traversal of graph.
  22. Write a function to determine whether or Not Graph G is an undirected graph.
  23. Write a procedure which returns the degree of a given node.
  24. Highlight the difference between DFS and BFS.
  25. Write a program to count the number of nodes in a graph.
  26. Write a program to count the indegree and outdegree of every node in a graph
  27. Write a program to find the successors and predecessors of each node in a graph.
  28. Solve exercise 9.4 from Robert Kruse “Data Structures and Program Design in C” .
  29. Solve exercises 11.7(a), (b) from Robert Kruse “Data Structures and Program Design in C” .

- 
30. Solve 7.4 from Robert Kruse “Data Structures and Program Design in C” .
  31. Solve 11.1 from Robert Kruse “Data Structures and Program Design in C” .
  32. Solve 11.2 from Robert Kruse “Data Structures and Program Design in C”
  33. Write a function to check whether a graph contains a cycle
  34. Write a function to check whether a given graph has Ring topology
  35. Function to check whether a given graph has Mesh topology
  36. Write a C Program for Find if there is a path between two vertices in a directed graph.
  37. Write a program in C to Count all possible paths between two vertices.
  38. Minimum number of edges between two vertices of a Graph