An adjacency list is maintained for each node present in graph which stores node value and a pointer to next adjacent node to respective node.

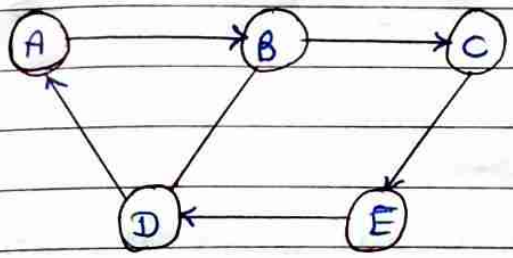


fig : Directed graph

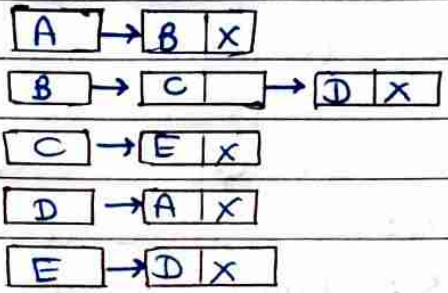


fig : Adjacy list

In directed graph, sum of lengths of all the adjacy lists is equal to the number of edges preseht in the graph.

Graph Traversal Algorithm :-

In this tutorial we will learn all techniques by using which, we can traverse all the vertices of the graph. Traversing means examining all nodes and vertices of graph. There are two standard methods by using which, we can traverse graphs.

- Breadth first search
- Depth first search

1. Breadth first search (BFS) algorithm :-

Breadth first search is a graph traversal algorithm that starts traversing graph from root node and explores all the neighbouring nodes.

Then, it selects nearest node and explore all unexplored nodes. The algorithm follows same process for each of nearest node until it finds goal.

Algorithm :-

step 1: SET STATUS = 1 (ready state) for each node in G.

step 2: Enqueue starting node A & set its STATUS = 2 (waiting state).

step 3: Repeat steps 4 and 5 until QUE E is empty.

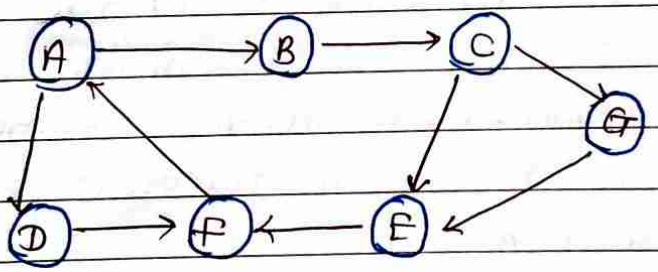
step 4: Dequeue a node N. Process it & set its STATUS = 3

step 5: Enqueue all neighbours of N that are in ready state (whose STATUS = 1) & set (STATUS = 2) [END OF LOOP].

step 6: EXIT.

Example :-

Consider graph G shown in following image, calculate minimum path P from node A to node E. Given that each edge has a length of 1.



Adjacency lists :

A : B, D
B : C, F
C : E, G
G : E
E : B, F
F : A
D : F.

Solution :-

Minimum Path P can be found by applying Breadth first search algorithm that will begin at node A and will end at E.

A → B → C → E

Depth first search Algorithm :-

DFS algorithm starts with initial node of graph G, & goes to deeper & deeper until we find goal node / node which has no children. The data structure used in DFS is stack.

Algorithm :-

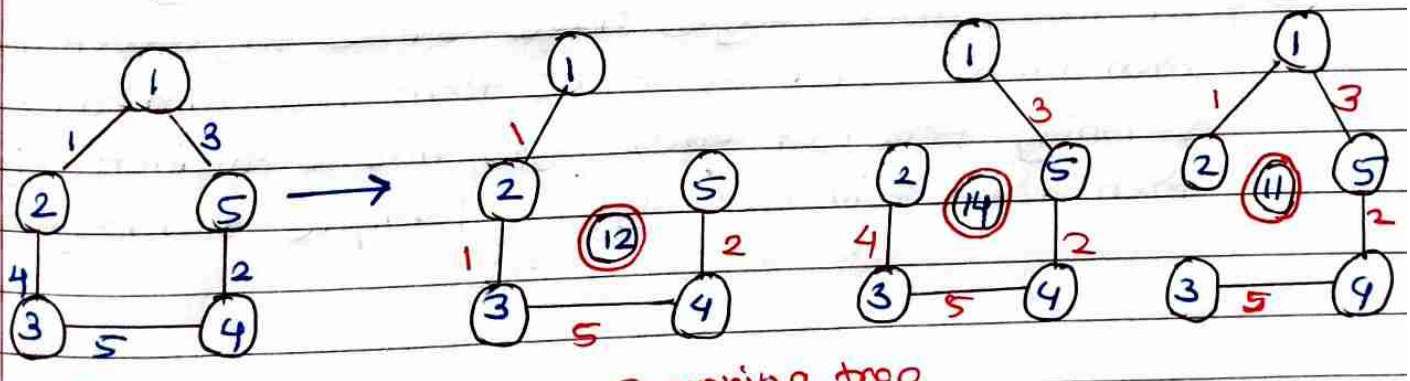
step 1: SET STATUS = 1 (ready state) for each node in G.
step 2: Push starting node A on stack & set its STATUS = 2 (waiting state).
step 3: Repeat steps 4 and 5 until stack is empty.
step 4: Pop top node N. Process it & set its STATUS = 3.
step 5: Push on stack all neighbours of N that are in ready state (whose STATUS = 1) and set their STATUS = 2 (waiting state) [END OF LOOP].
step 6: EXIT.

Spanning Tree :-

If we have a graph containing V vertices and E edges, then graph can be represented as: G(V, E). If we create spanning tree from above graph, then spanning tree would have same number of vertices as the graph, but vertices are not equal.

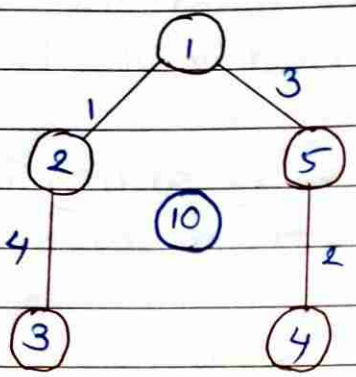
edges (Spanning tree) = no. of edge (in graph) - 1.

Example :-



Minimum Spanning Trees :-

The minimum spanning tree is a tree whose sum of edge weights is minimum.



In above tree, total edge weight is less than above spanning trees, therefore a minimum spanning tree is a tree which is having an edge weight i.e. 10.

Properties of Spanning tree :-

- A connected graph can contain more than one spanning tree.
- All possible spanning trees that can be created from given graph G would have same number of vertices in given graph minus 1.
- Spanning tree does not contain any cycle, let's understand this property through an example.
- Spanning tree cannot be disconnected. If we remove one more edge from any of above spanning trees as.
- If two / more edges have same edge weight, then there would be more than two minimum spanning tree. If each edge has a distinct weight, then there will be only one / unique spanning tree.

Applications of Spanning tree :-

1). Building a network :- Suppose there are many routers in network connected to each other, so there might be a possibility that it forms a loop.

2). Clustering :- clustering means that grouping set of objects in such way that similar objects belong to same group than to different group. our goal is to divide the n objects into k groups such that distance between different groups gets maximised.

Searching :-

Searching is a process of finding some particular element in list. If the element is present in the list, then process is called successful and process returns location of that element, otherwise search is called unsuccessful.

There are two methods widely used as below:

- Linear search
- Binary search

1). Linear search :-

linear search is a simplest sequential search algorithm and often called sequential search. In this type of searching, we simply traverse the list completely and match each element of list with item whose location is to be found.

linear search is mostly used to search an unordered list in which items are not sorted. The algorithm is given as follows :