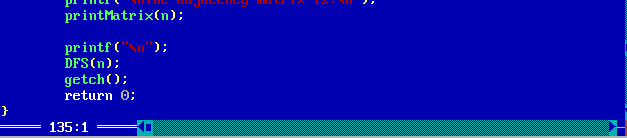
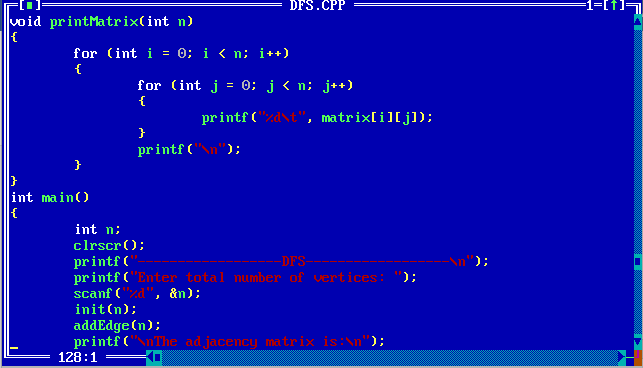
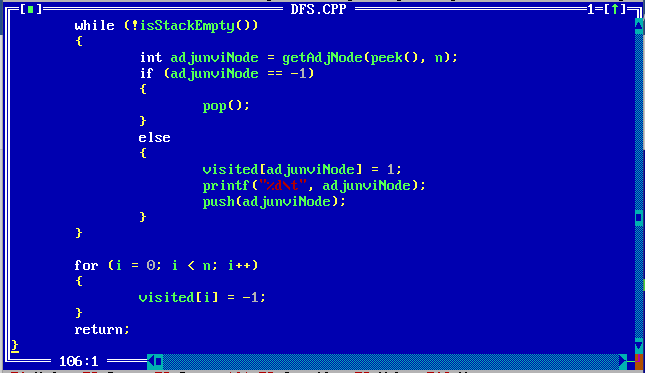
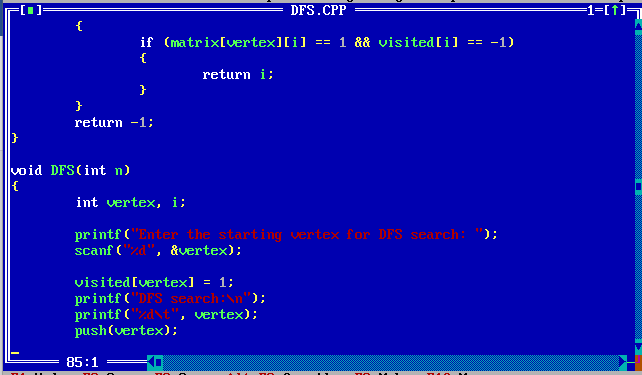
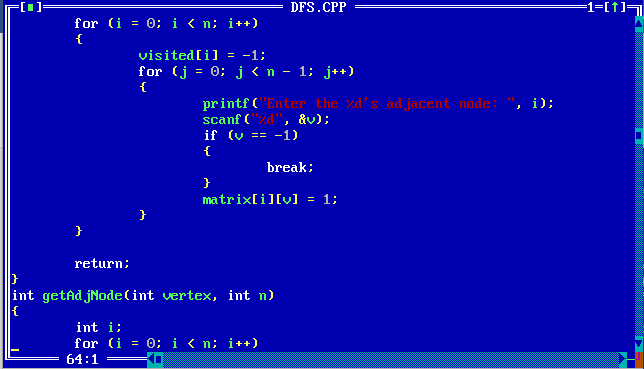
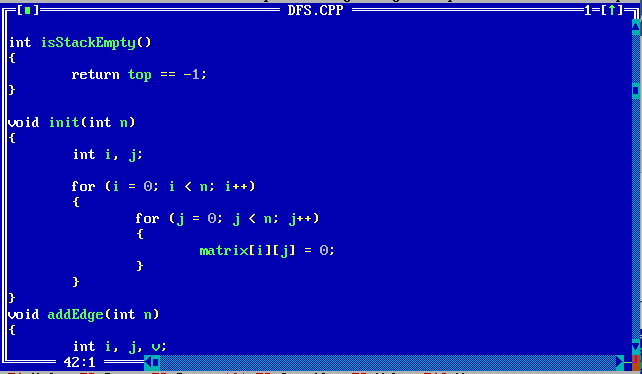
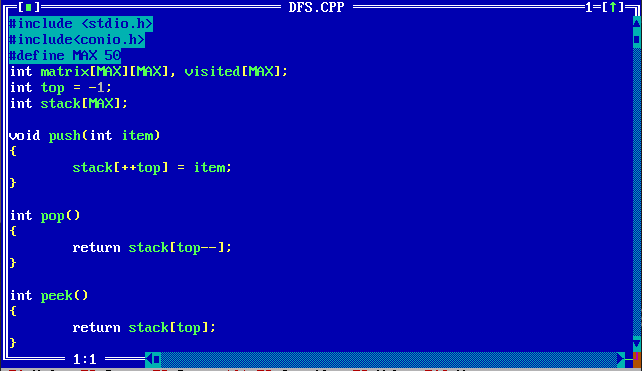
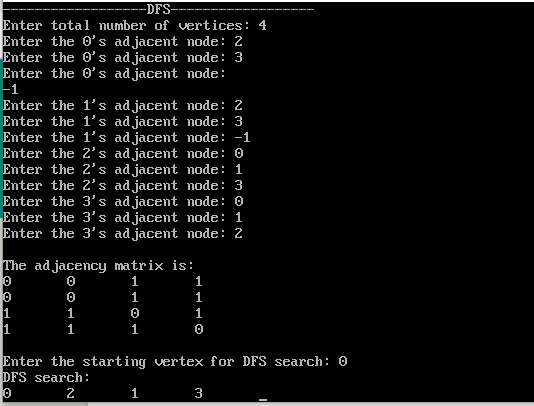
1)DFS using adjacency matrix(iterative):

Code:



Output:  


Time Complexity Analysis:

+ Functions init , addEdge and printMatrix- 2 nested loops => O(n^2)

+ Function DFS- 2 loops =>n+n=2n=>O(n)

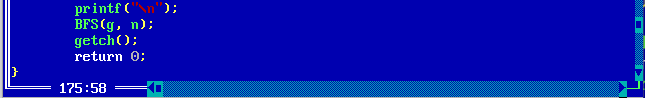
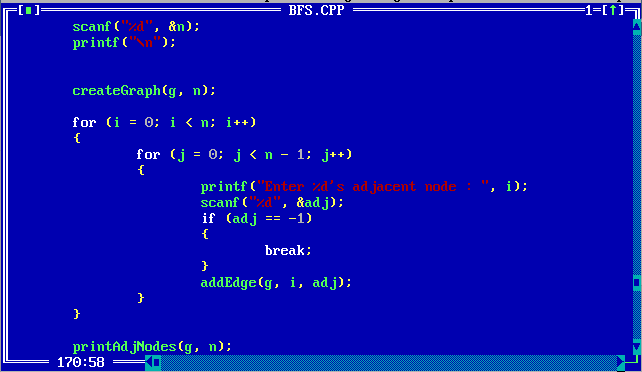
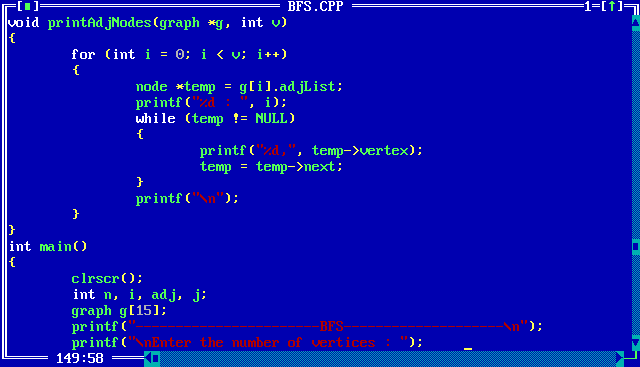
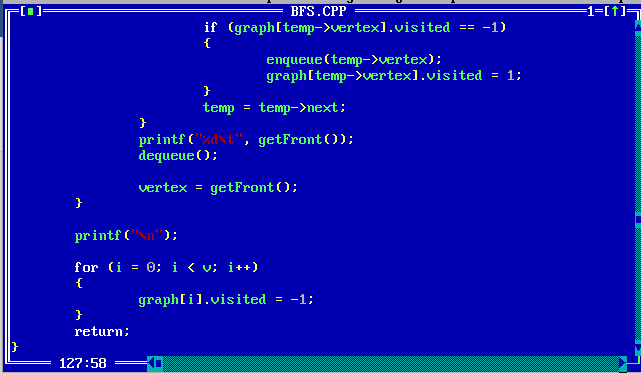
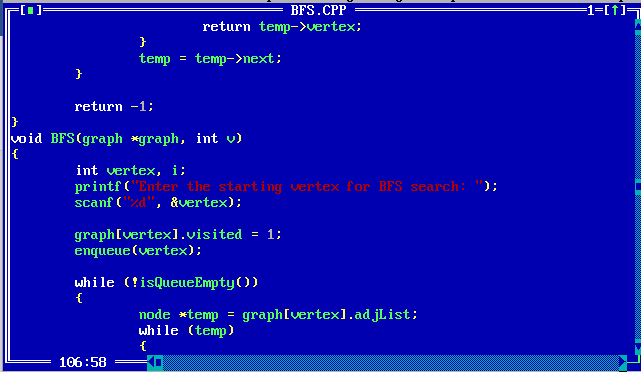
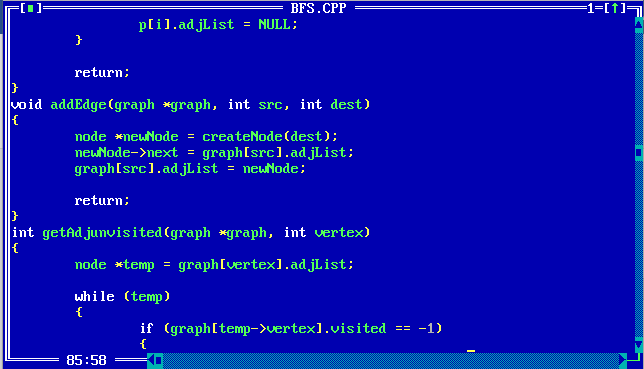
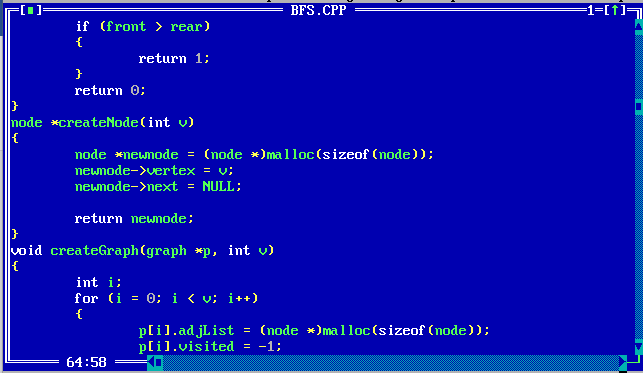
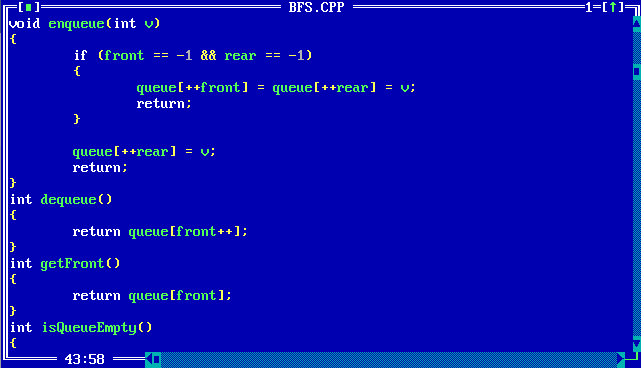
Let V be the number of vertices and E be the number of edges.

We are using adjacency matrix here.During dfs traversal,each vertex v is called atleast once.From each vertex, all the vertices are checked until a negihbouring vertex is found.The complexity of this proces is O(V).Since this process is done for all the vertices,The Overall complexity is O(VxV)=O(V2).

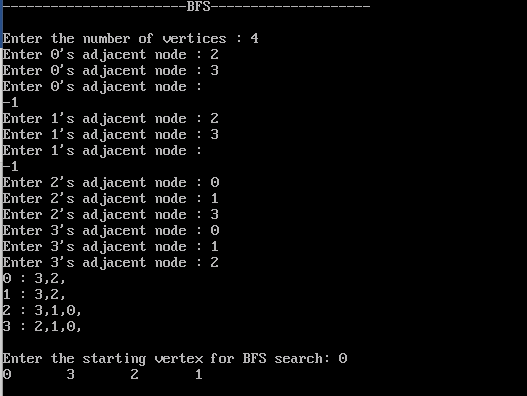
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

2)BFS using adjacency list(iterative):

Code:



Output:



Time Complexity :

+ Function printAdjNodes, getAdjunVisited and createGraph – one loop => O(n)

+Function BFS – Two loops=>n+n=2n=>O(n)

Let V be the number of vertices and E be the number of edges.

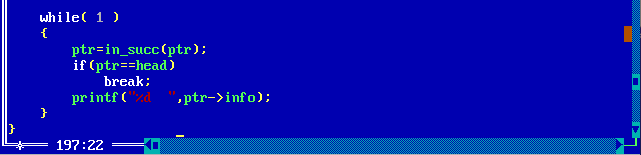
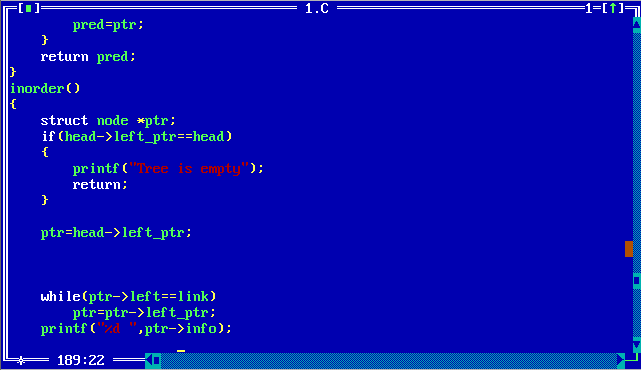
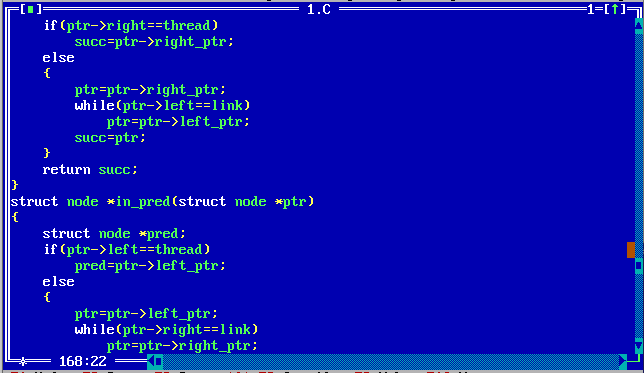
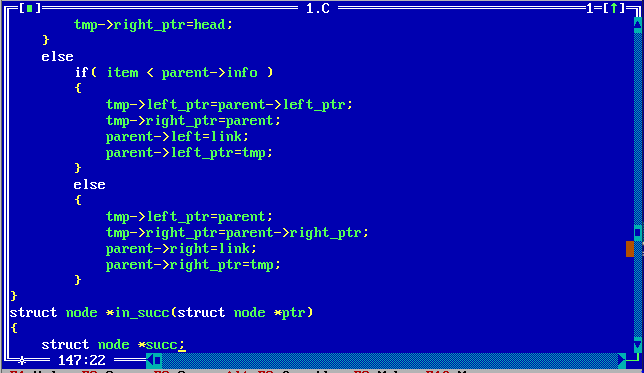
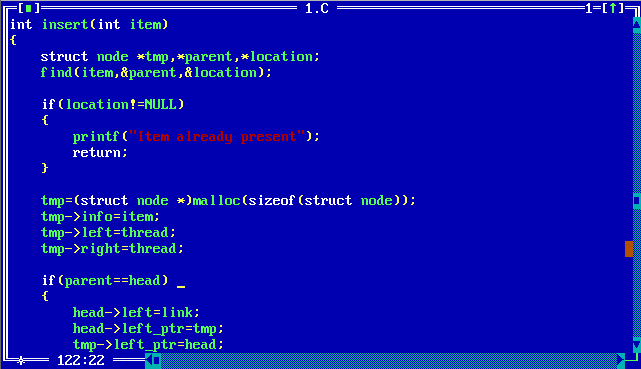
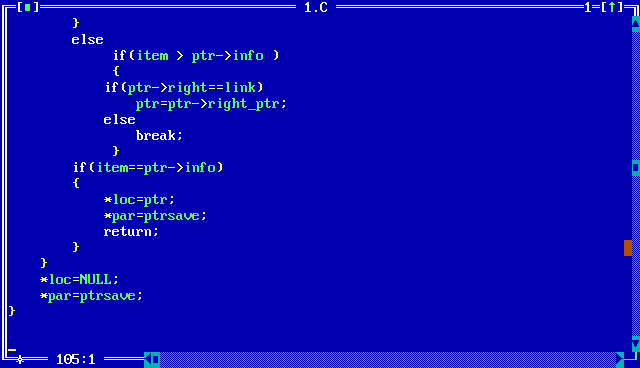
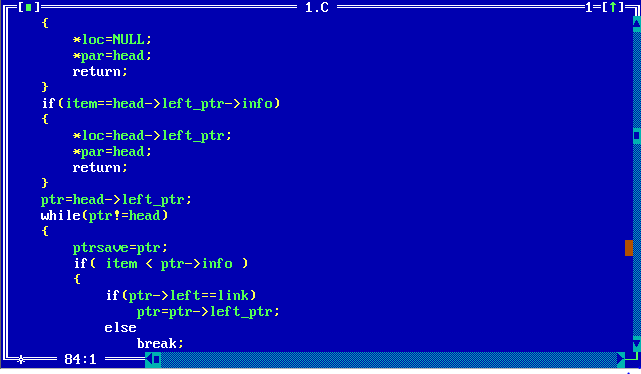
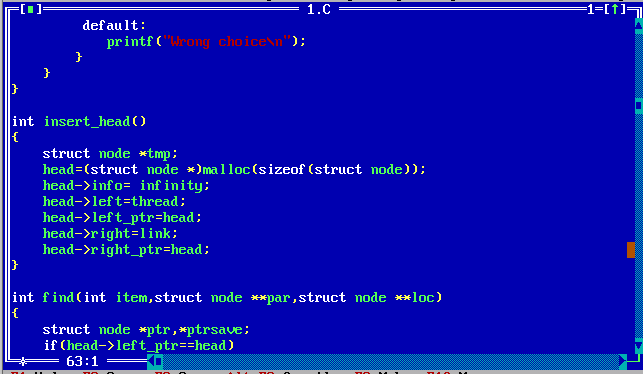
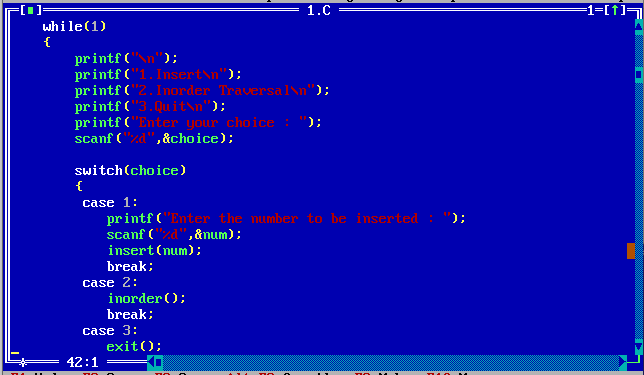
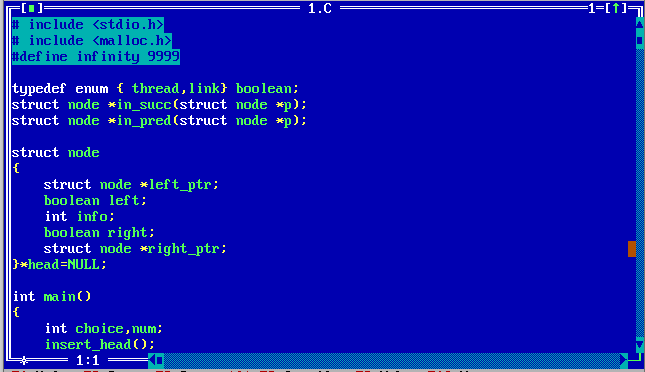
We are using adjacency list here.During bfs traversal,each vertex v is called atleast once.From,each vertex all of its neighbouring vertices are checked,it’s complexity (summing over all vertices) is O(E).Constant operation of changing the visited flag to 1 happens for every vertex.It’s complexity is O(V).The overall complexity is O(V)+O(E)=O(V+E).

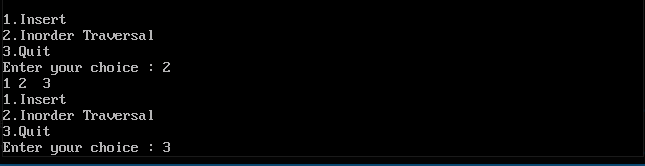
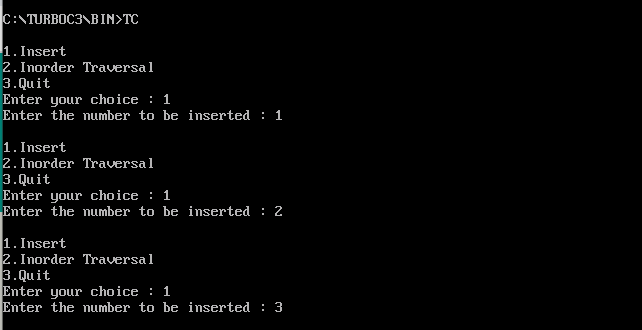
That is the complexity is O(n)

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

3)Threaded Binary tree:

Code:



Output: 

Time Complexity:

Functions find , inorder- loop =>O(n)

Other functions have complexities which are constants , let us assume it to be x

Therefore the total time complexity of this program is O(n)

Aadharsh R

2019103604

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

[Join Microsoft Teams Meeting](https://teams.microsoft.com/l/meetup-join/19%3aebf7ffcc81634f3d93fa3db40fc984c6%40thread.tacv2/1601262783075?context=%7b%22Tid%22%3a%226e804f24-0209-4dcd-ac89-97525eddbd30%22%2c%22Oid%22%3a%226e095e16-c2d7-4e55-88f2-af6be74065ed%22%7d)

[Learn more about Teams](https://aka.ms/JoinTeamsMeeting)