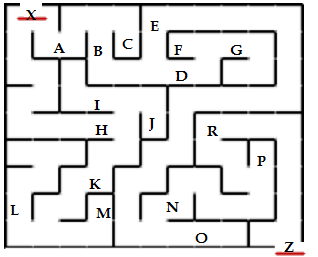
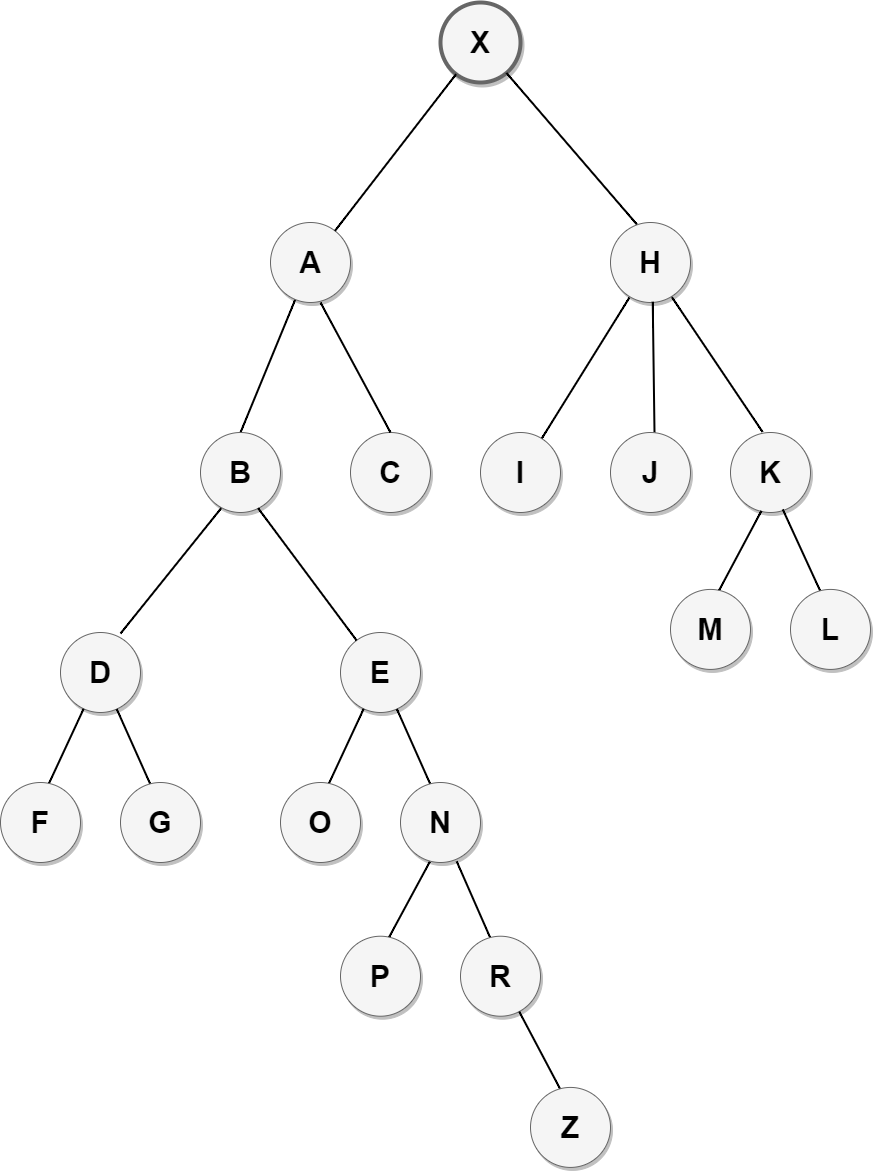
Buğra ATEŞ

Maze Solving with

Depth First Search Algorithm in C



1. Create the graph



1. Create an adjacency matrix for your graph

A B C DE F GH I J K LMNO PR X Z

 matrix.txt file for reading

1. Using this graph solve the maze using DFS algorithm. Your algorithm will start from the root element X and should reach the end element Z… Use C language for your algorithm.

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

int Array[19][19];

char letters[19] = {'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','R','X','Z'};

bool visited[19];

int countStep = 1;

void DFS(int root, bool visited[])

{

int i;

visited[root] = true;

if(letters[root] == 'X')

{

printf("\n\n %d. \tStep | Index : %d \tLetter : %c ==> ROOT ", countStep,root,letters[root]);

countStep++ ;

}

else

{

printf("\n\n %d. \tStep | Index : %d \tLetter : %c ", countStep,root,letters[root]);

countStep++ ;

};

for( i = 0; i < 19; i++)

{

if(Array[root][i] == 1 && visited[i] == false)

{

DFS(i, visited);

}

} if(letters[root] == 'Z') {

printf("\n\n\n Maze solved in %d steps (%c), reached to exit point.",countStep-1,letters[root]);

printf("\n\n DFS Algorithm is continuing to travel the remaining nodes...\n"); } }

void readMatrix()

{

int i = 0;

FILE \*fp = fopen("matrix.txt", "r");

while(fscanf(fp, "%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d",

&Array[i][0],

&Array[i][1],

&Array[i][2],

&Array[i][3],

&Array[i][4],

&Array[i][5],

&Array[i][6],

&Array[i][7],

&Array[i][8],

&Array[i][9],

&Array[i][10],

&Array[i][11],

&Array[i][12],

&Array[i][13],

&Array[i][14],

&Array[i][15],

&Array[i][16],

&Array[i][17],

&Array[i][18]

) != EOF)

{

i = i + 1 ; }}

**i**nt main()

{

for(int i=0 ; i<19 ;i++)

{

visited[i] = false ;

}

readMatrix();

DFS(17, visited);

return 0;

}

This program shows every step of nodes and the solution of maze.

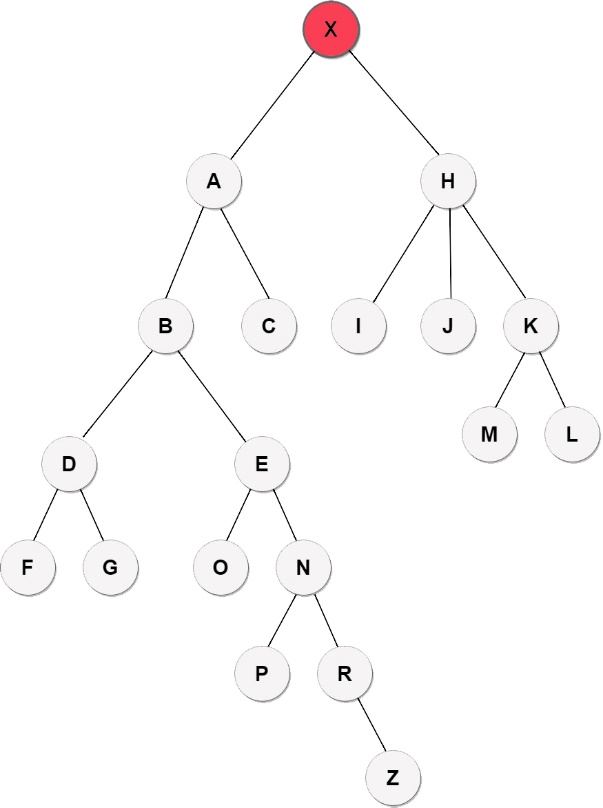
X = ROOT = 17

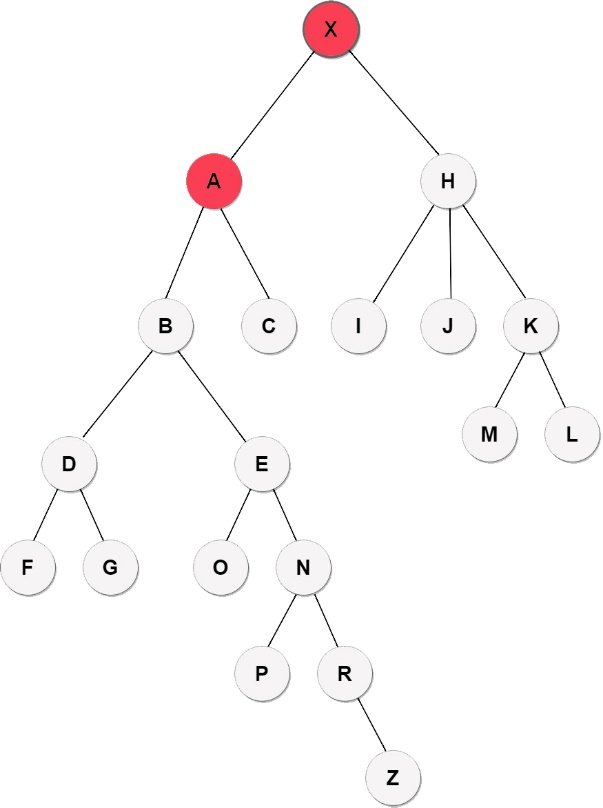
Program files

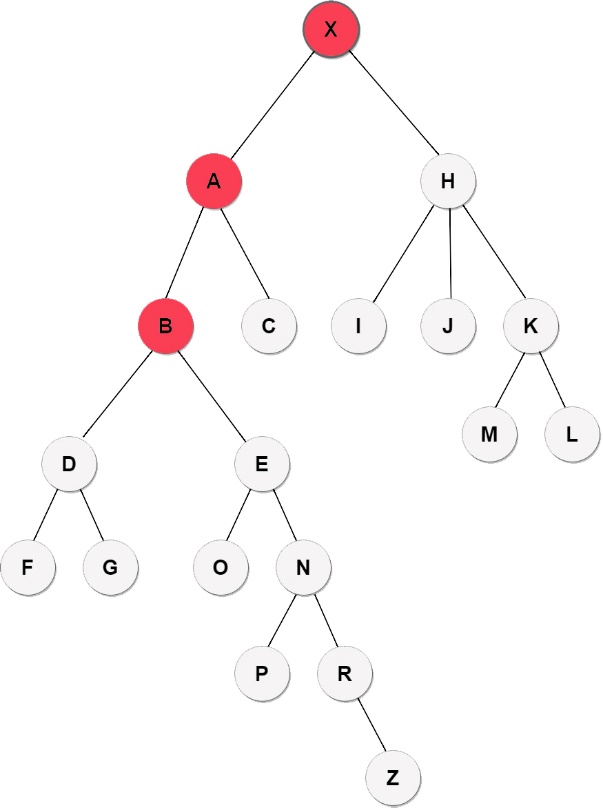
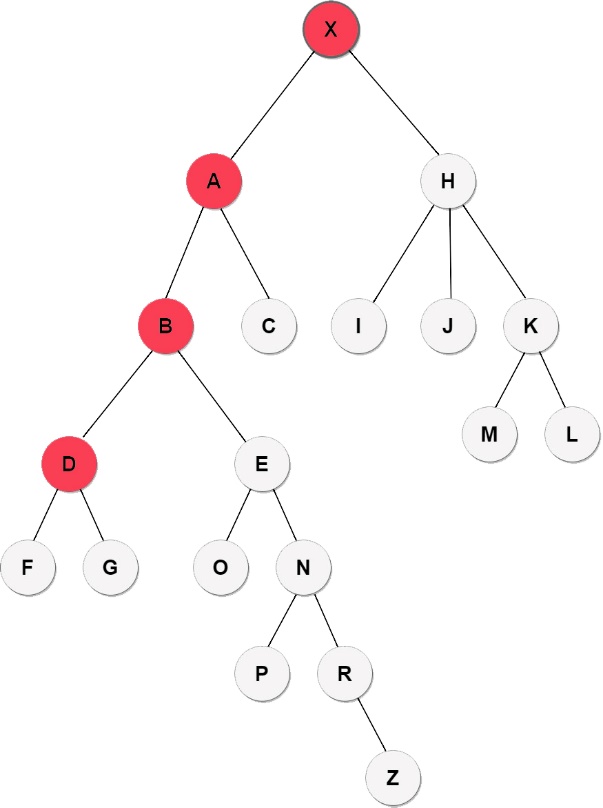
****

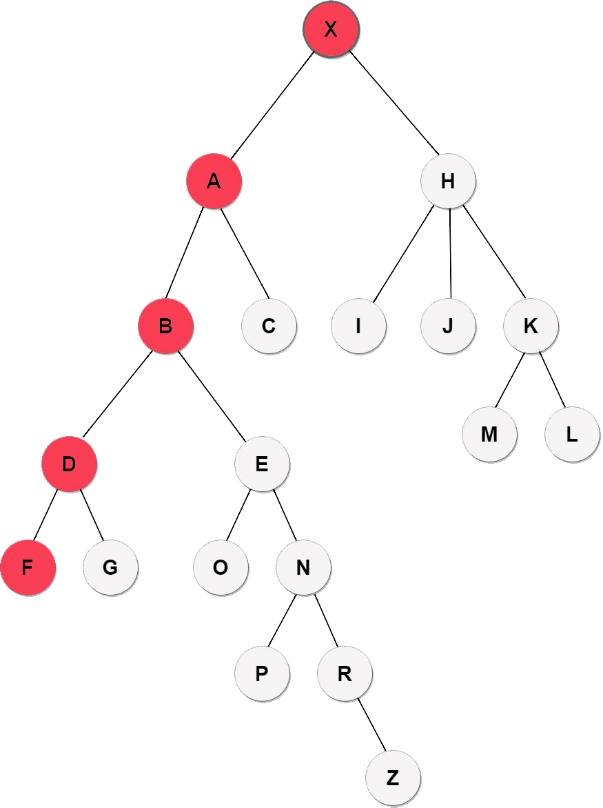
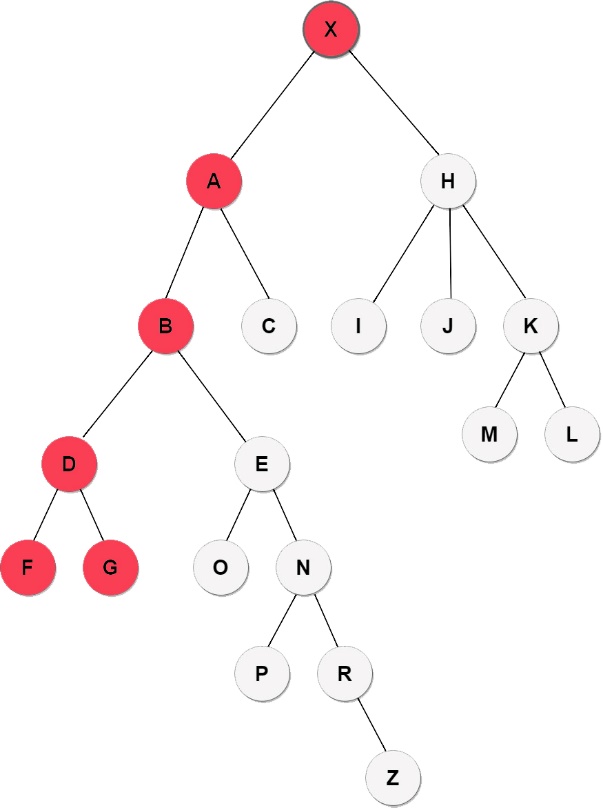
1. Trace the path of your algorithm… For each step which node is selected? Write the nodes…

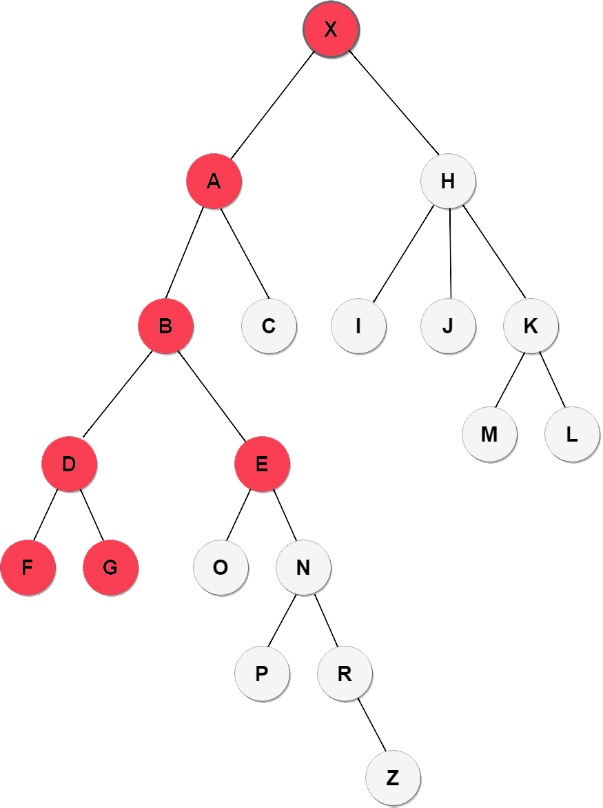
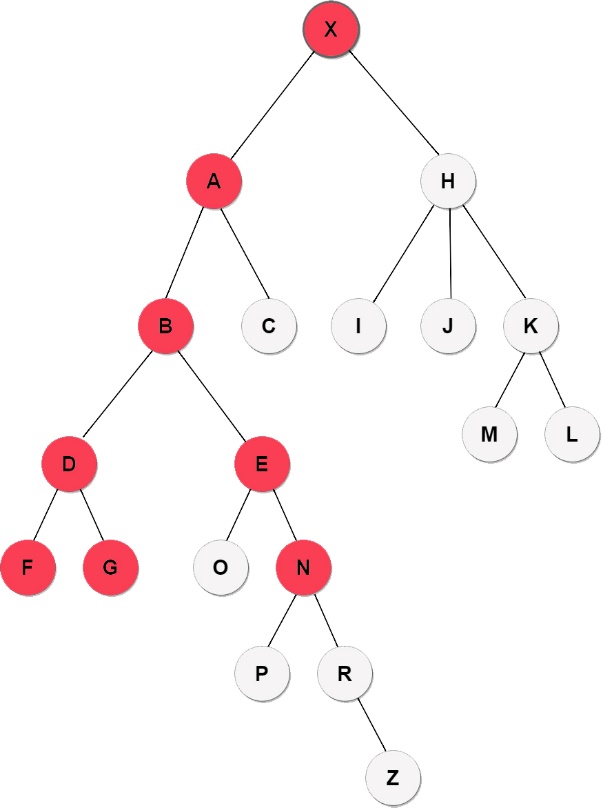
GRAPH VISULATION

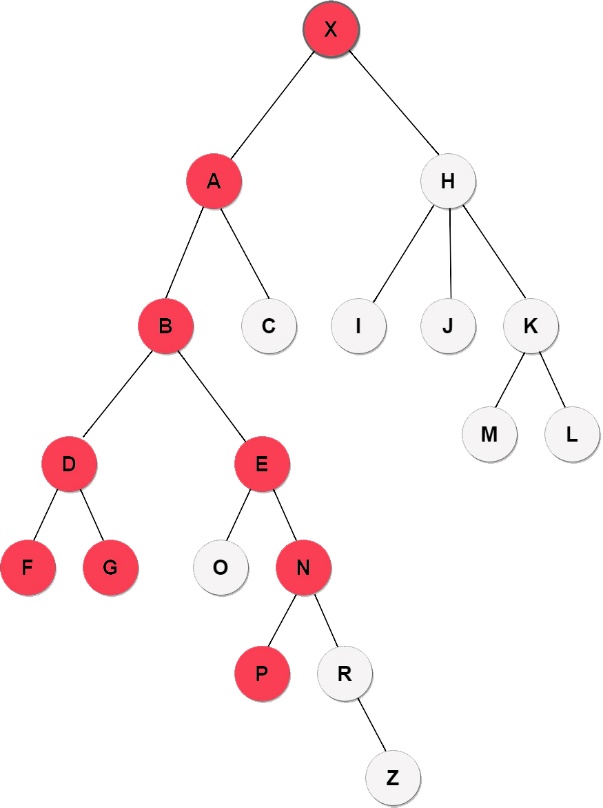
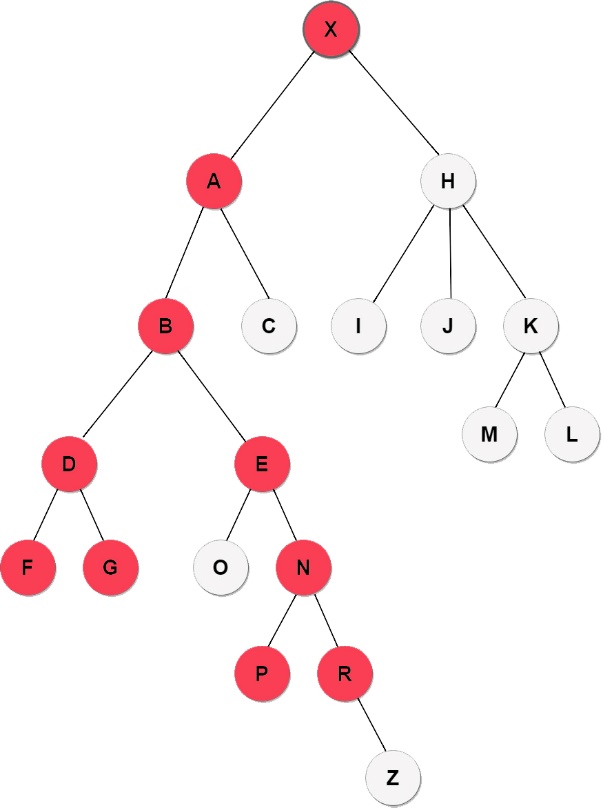


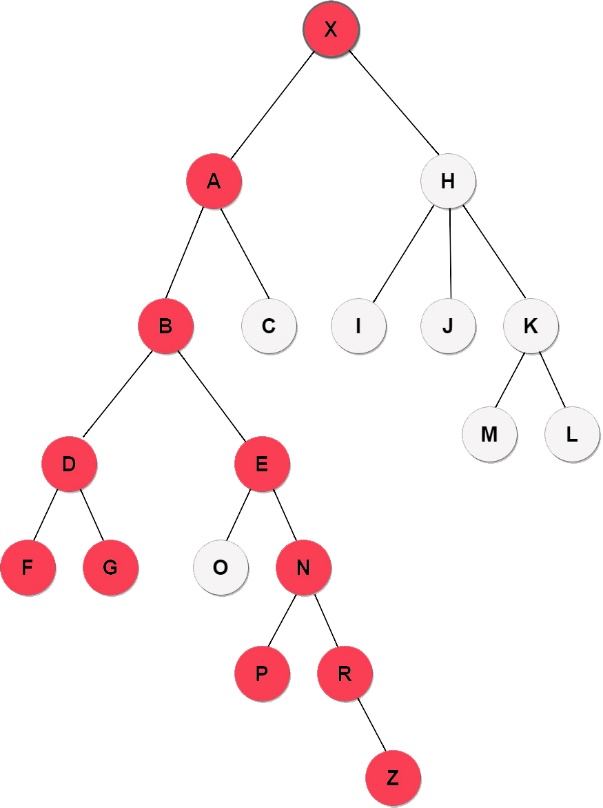
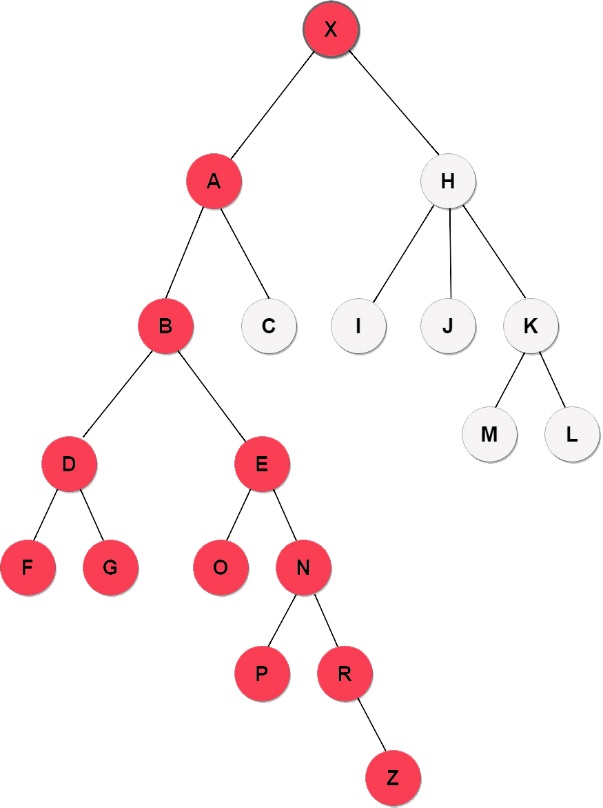






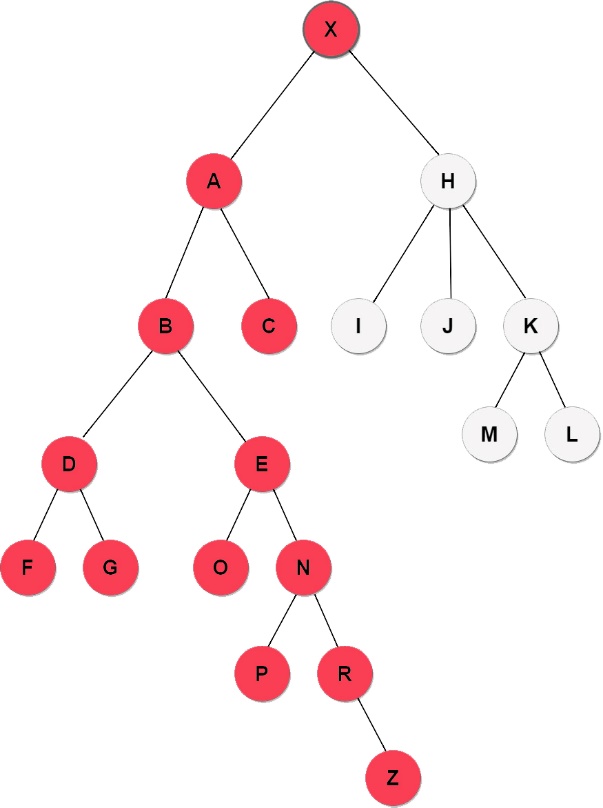
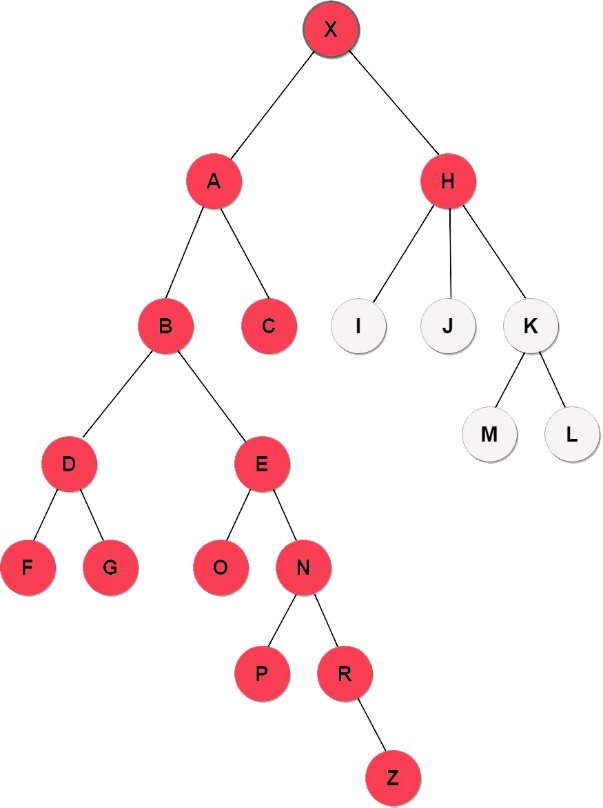


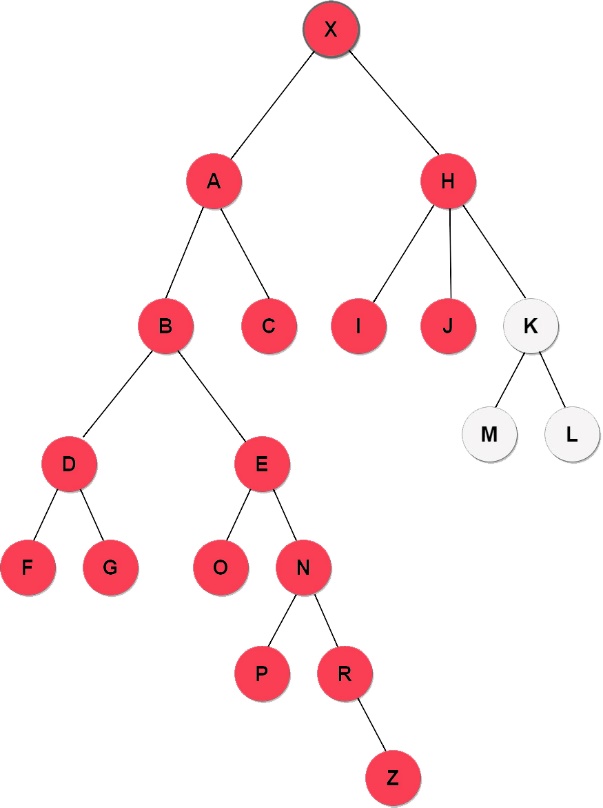
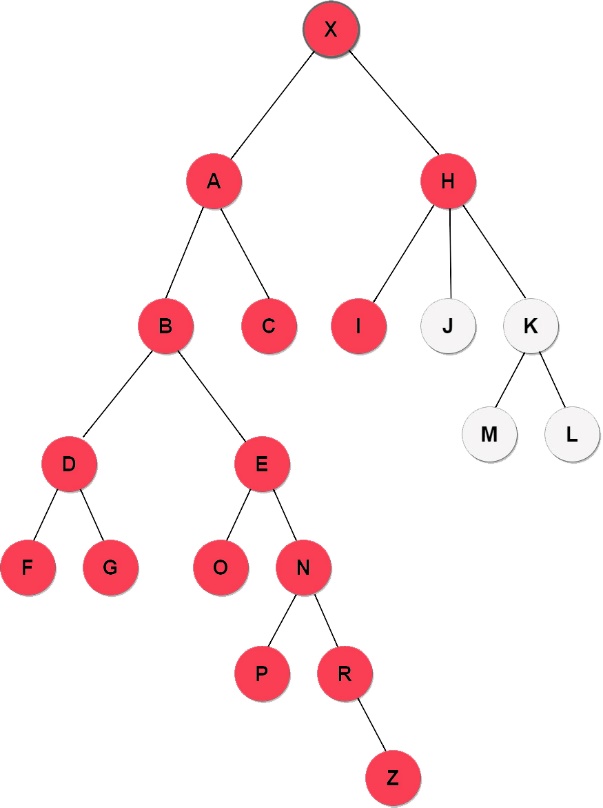


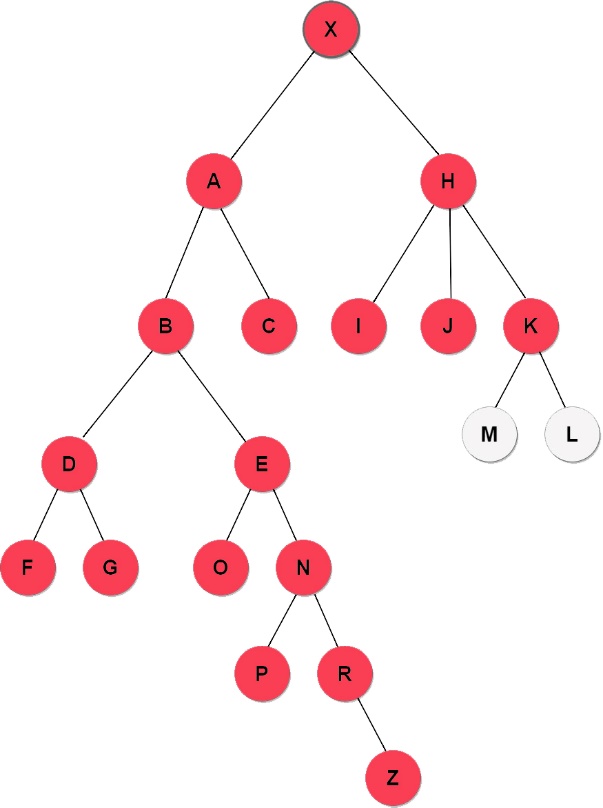
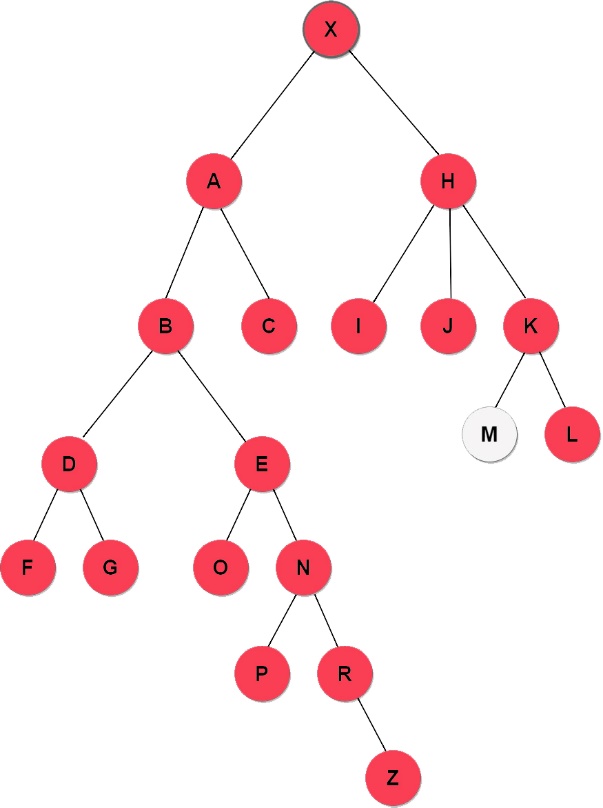


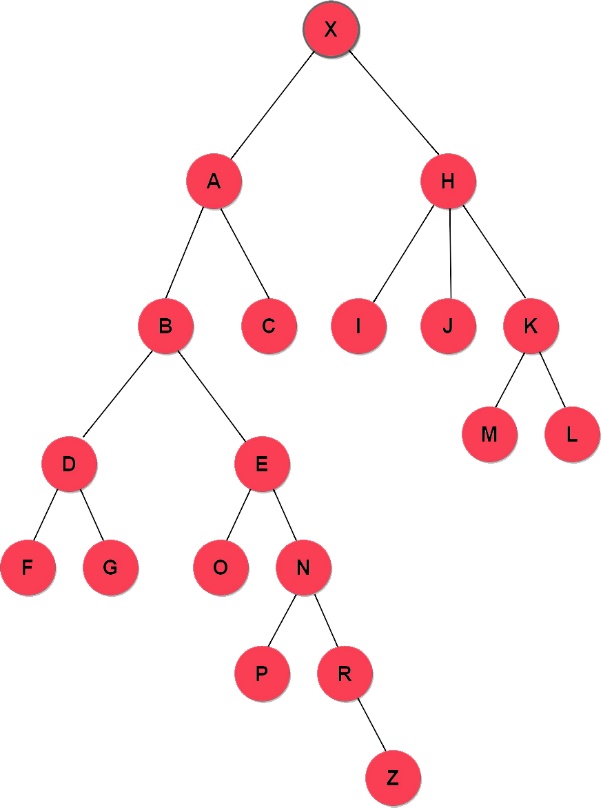
MAZE SOLVED Continuing to finish travelling

Program shows the exit point (Z)









Program ends, all nodes travelled.

1. Step | Index : 17 Letter : X ==> ROOT

2. Step | Index : 0 Letter : A

3. Step | Index : 1 Letter : B

4. Step | Index : 3 Letter : D

5. Step | Index : 5 Letter : F

6. Step | Index : 6 Letter : G

7. Step | Index : 4 Letter : E

8. Step | Index : 13 Letter : N

9. Step | Index : 15 Letter : P

10. Step | Index : 16 Letter : R

11. Step | Index : 18 Letter : Z

Maze solved in 11 steps (Z), reached to exit point.

DFS Algorithm is continuing to travel the remaining nodes...

12. Step | Index : 14 Letter : O

13. Step | Index : 2 Letter : C

14. Step | Index : 7 Letter : H

15. Step | Index : 8 Letter : I

16. Step | Index : 9 Letter : J

17. Step | Index : 10 Letter : K

18. Step | Index : 11 Letter : L

19. Step | Index : 12 Letter : M

Program end, all nodes travelled.

Buğra ATEŞ

B1705.090046