**DESIGN AND ANALYSIS OF ALGORITHMS**

PROJECT

on

**RAT IN A MAZE USING BACKTRACKING**

Made by:

Pulkit Vaid 169108109

Samridhi Kohli 169108125

**CODE**

#include<stdio.h>

#include<stdbool.h>

int N;

bool isSafe(int maze[N][N],int x,int y)

{

if(x>=0&&x<N&&y>=0&&y<N&&maze[x][y]==1)

return true;

return false;

}

bool solveMazeUtil(int maze[N][N],int x,int y,int sol[N][N])

{

if(x==N-1&&y==N-1)

{

sol[x][y]=1;

return true;

}

if(isSafe(maze,x,y)==true)

{

sol[x][y]=1;

if (solveMazeUtil(maze,x+1,y,sol)==true)

return true;

if (solveMazeUtil(maze,x,y+1,sol)==true)

return true;

sol[x][y]=0;

return false;

}

return false;

}

void printSolution(int sol[N][N])

{

int i,j;

printf("The rat can take the following path indicated by 1\n");

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

{

for(j=0;j<N;j++)

printf(" %d ", sol[i][j]);

printf("\n");

}

}

bool solveMaze(int maze[N][N])

{

int sol[N][N];

int i,j;

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

{

for(j=0;j<N;j++)

{

sol[i][j]=0;

}

}

if(solveMazeUtil(maze, 0, 0, sol) == false)

{

printf("Solution doesn't exist");

return false;

}

printSolution(sol);

return true;

}

int main()

{

int i,j;

printf("Enter the size of your matrix:\n");

scanf("%d",&N);

printf("Enter the matrix 1 for unblocked and 0 for blocked:\n");

int maze[N][N];

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

{

for(j=0;j<N;j++)

{

scanf("%d",&maze[i][j]);

}

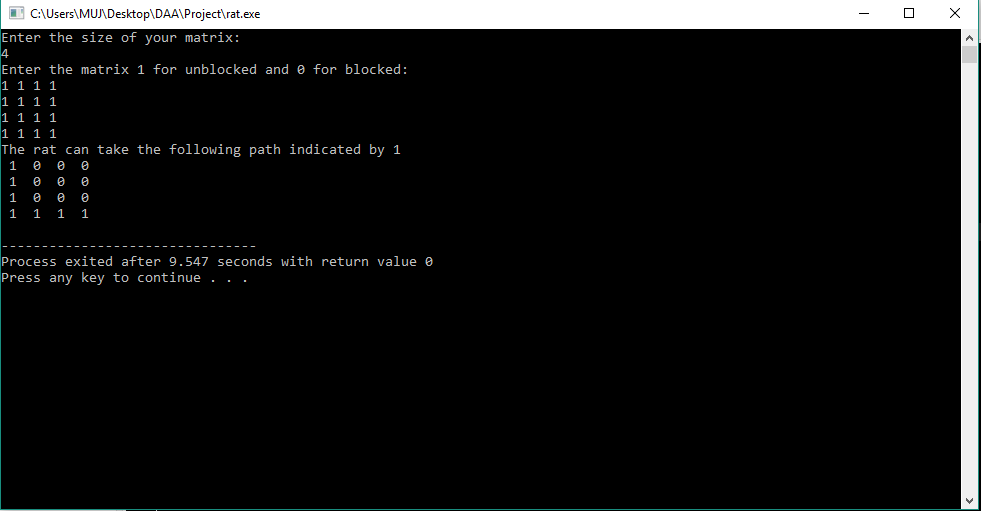
}

solveMaze(maze);

return 0;

}

**OUTPUT**



**TIME AND SPACE COMPLEXITY**

Time complexity is 2^n\*m and space complexity is 0(n\*m)