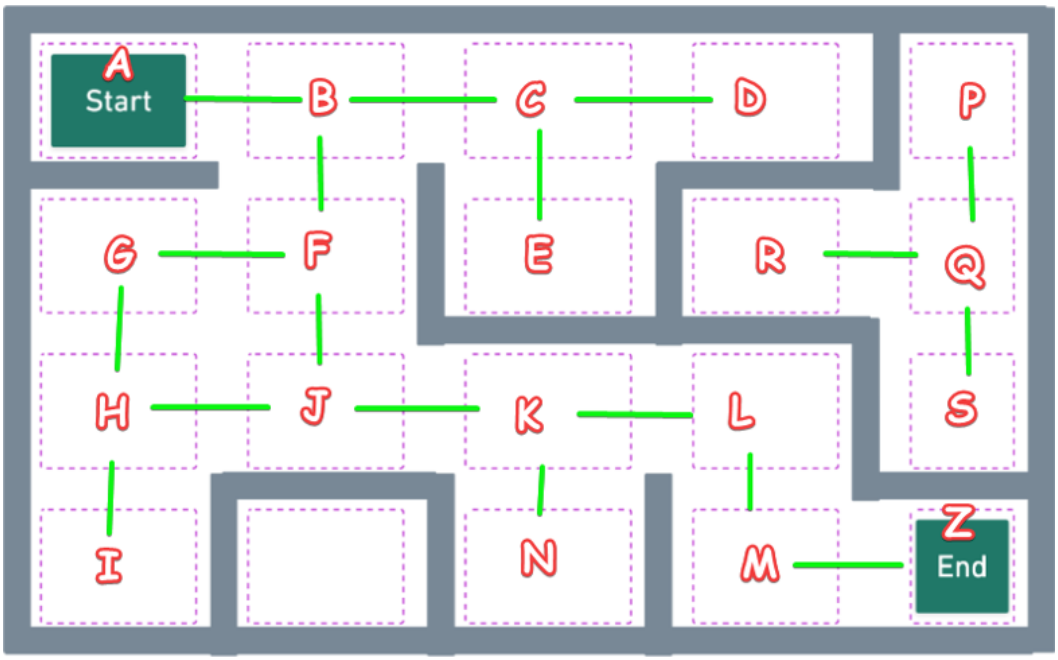
Artificial Intelligence (CS308)

Assignment - 7

U19CS012

Maze Problem

You are Given a Maze of Room which are Labelled with Capital Letters like 'A' & 'B'.



Input:

Source Room and **Destination** Room.

Output:

To Find out **Where the Destination Room can be Reached or Not?**

If it can be Reached, Print any Path to reach it.

PROLOG Code {Using B.F.S.}

```
% path('X','Y') -> There is Path from Point 'X' to Point 'Y'
path('A','B').

path('B','C').
path('B','F').

path('C','D').
path('C','E').

path('F','G').
path('F','J').

path('G','H').

path('H','I').
path('H','J').

path('J','K').

path('K','L').
path('K','N').

path('L','M').
path('M','Z').

path('P','Q').

path('Q','R').
path('Q','S').

% Prolog F(x) to Append 2 List's
appnd([],X,X).
appnd([H|T],N,[H|T1]):-
    appnd(T,N,T1).

% F(x) to extract all the Un-Visited Nodes in Graph
extend([Node|Path],NewPaths):-
    bagof([NewNode,Node|Path],
          ( (path(Node,NewNode);path(NewNode,Node)),
            not(member(NewNode,[Node|Path]))),
            NewPaths
         ),!.
    extend(_, []).

% Termination Condition If Our Final Node {Goal} is Reached
bfs([[Node|Path]|_],[Node|Path],Goal):-
    Node=Goal.
```

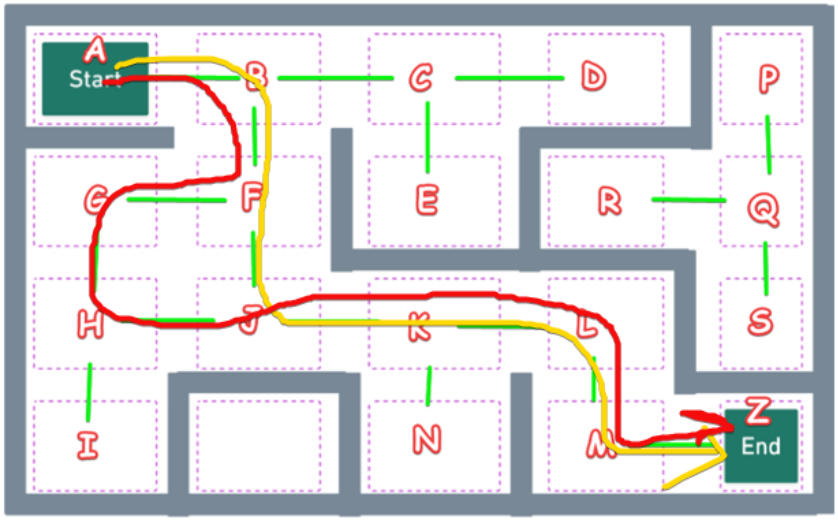
```
% B.F.S. Recursion
bfs([Path|Paths], Sol, Goal):-
    extend(Path, NewPaths),
    appnd(Paths, NewPaths, Paths1),
    bfs(Paths1, Sol, Goal).

% F(x) to Print the Entire Path from Source to Destination
displaypath([]):-
    write("END"),nl.

displaypath([H|T]):-
    write(H),
    write(" -> "),
    displaypath(T).

% Main F(x) to Intiate the Search
findpath(Start,Goal):-
    bfs([[Start]], Sol, Goal),
    reverse(Sol, Path),
    displaypath(Path).
```

Output

Q) *Does there a Path Exist from Point 'A' to Point 'Z'?*

```
?- findpath('A','Z').
A -> B -> F -> J -> K -> L -> M -> Z -> END
true ;
A -> B -> F -> G -> H -> J -> K -> L -> M -> Z -> END
true ;
false.
```

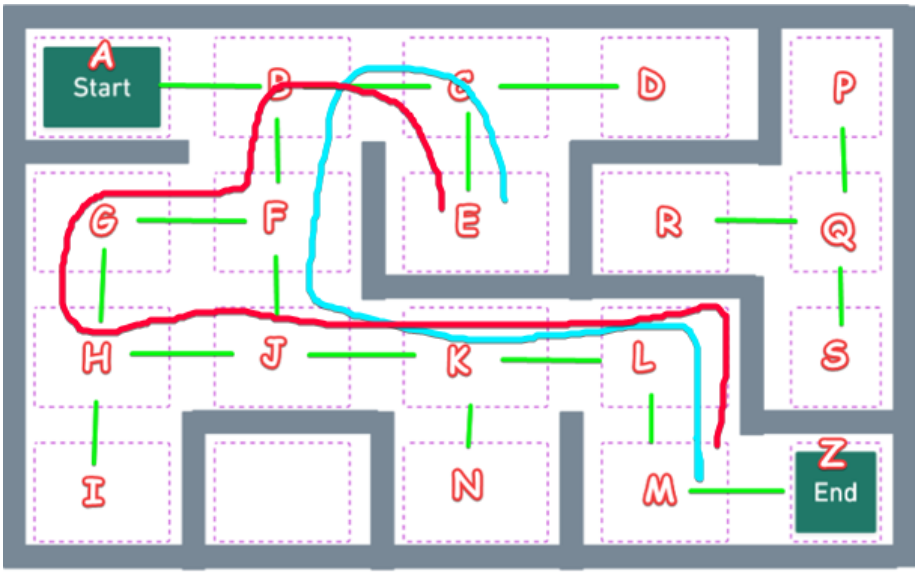
2 Paths to Reach 'Z'



[Surprisingly, There are **Two** Paths to Reach 'Z'.]

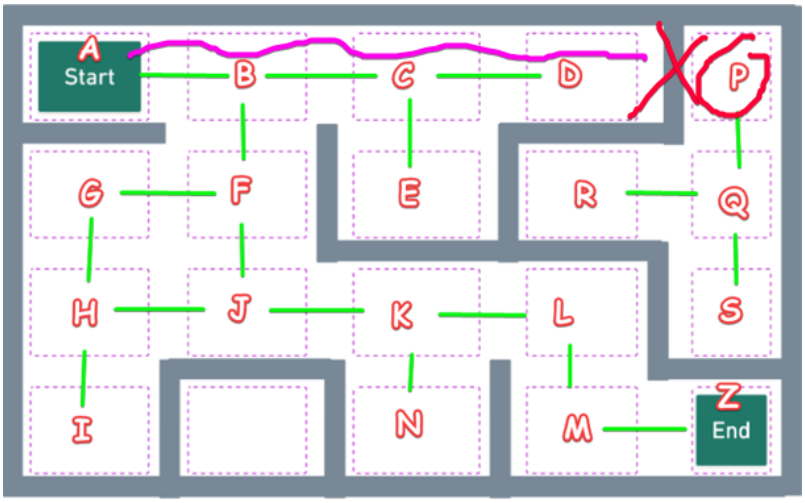
Q) *Does there a Path Exist from Point 'E' to Point 'M'?*

```
?- findpath('E','M').
E -> C -> B -> F -> J -> K -> L -> M -> END
true ;
E -> C -> B -> F -> G -> H -> J -> K -> L -> M -> END
true ;
false.
```



Q) *Does there a Path Exist from Point 'A' to Point 'P'?*

```
?- findpath('A','P').
false.
```



SUBMITTED BY: U19CS012

BHAGYA VINOD RANA