AI Assignment 2

(Ritisha Gupta, MT22056)

Ques: Search a road route from any city to any other city

It is essential to retrieve all the facts from the csv file names ‘roaddistance.csv’ as before that we can do anything. Using a prolog program, we will derive all the facts and store in ‘facts.pl’ file. The program will iterate through all rows and columns and then assert the facts accordingly. To insert into the we will use:

```
insert: -
    tell('c:/Users/HP/Desktop/AI_Ass2/facts.pl'),
    listing(distance),
    told.
```

To run use:

```
?-
%   library(record) compiled into record 0.00 sec, 71 clauses
%  library(csv) compiled into csv 0.03 sec, 179 clauses
%  library(lists) compiled into lists 0.00 sec, 0 clauses
% c:/Users/HP/Desktop/AI_Ass2/extract.pl compiled 0.05 sec, 8 clauses
?- csv('c:/Users/HP/Desktop/AI_Ass2/roaddistance.csv').
true .

?- save.
true.

?-
```

The facts.pl files look like.

```
distance('Agartala', 'Ahmedabad', 3305).
distance('Agartala', 'Bangalore', 3824).
distance('Agartala', 'Bhubaneshwar', 2286).
distance('Agartala', 'Bombay', 3593).
distance('Agartala', 'Calcutta', 1863).
distance('Agartala', 'Chandigarh', 2998).
distance('Agartala', 'Cochin', 4304).
distance('Agartala', 'Delhi', 2708).
distance('Agartala', 'Hyderabad', 3330).
distance('Agartala', 'Indore', 2891).
distance('Agartala', 'Jaipur', 2801).
distance('Agartala', 'Kanpur', 2281).
distance('Agartala', 'Lucknow', 2252).
distance('Agartala', 'Madras', 3493).
distance('Agartala', 'Nagpur', 2696).
```

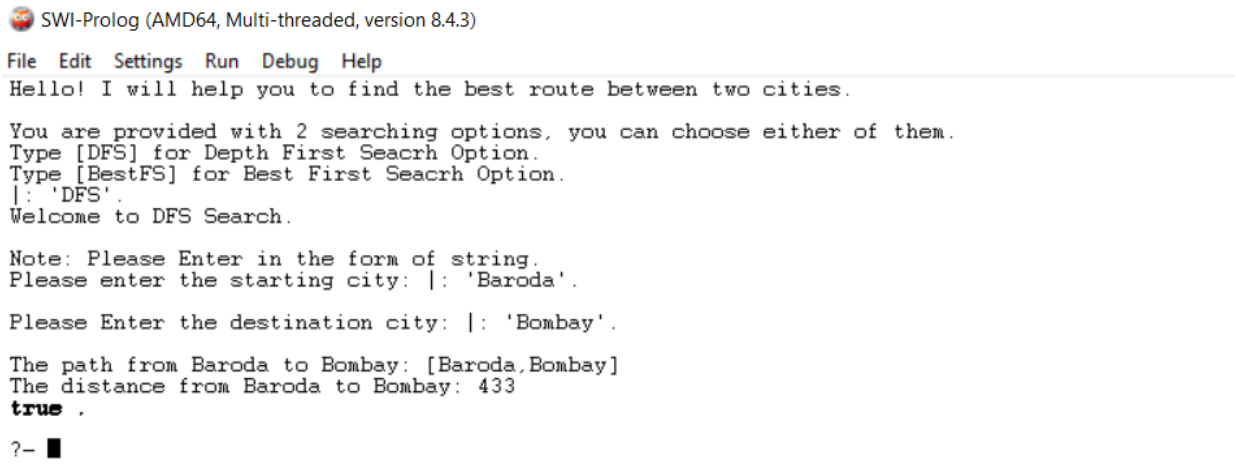
I am opting for option (a) i.e. Depth First Search and Best First Search.

Depth First Search.

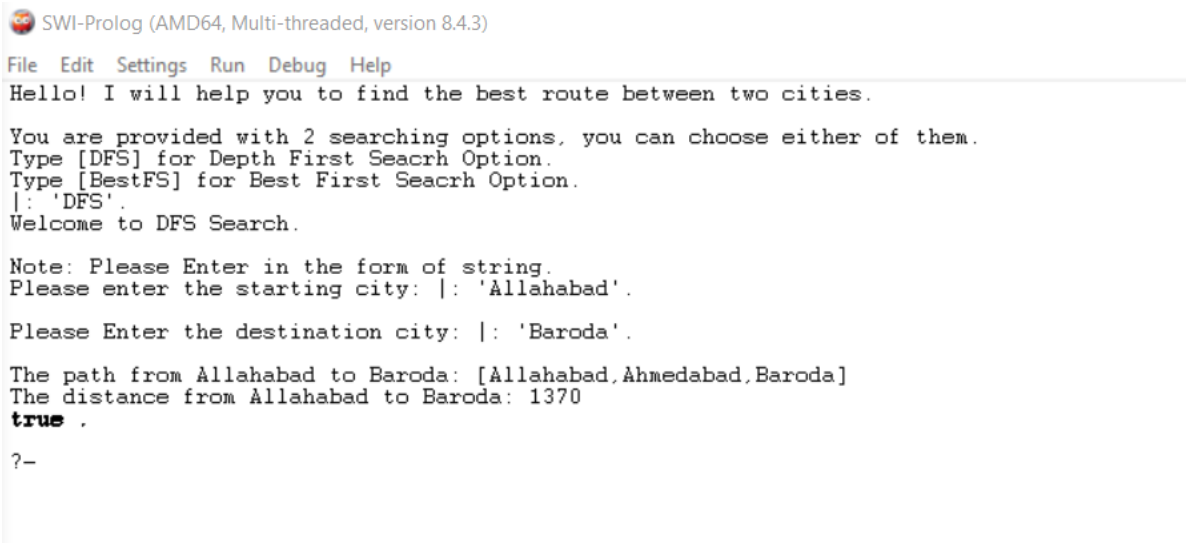
```
% Depth First Search
dfs(Src,Dest,[Src|Next],Cost):-
    connected(Src,Dest,Next,Cost).

% direct path rule
connected(Src,Dest,[Dest],Cost):-
    dist(Src,Dest,Cost).

% indirect path rule
connected(Src,Dest,[Mid|L],Cost):-
    dist(Src,Mid,Cost1),
    connected(Mid,Dest,L,Cost2),
    Cost is Cost1 + Cost2.



% to handle bidirectionality of roads
dist(X,Y,D):- distance(X,Y,D).
dist(X,Y,D):- distance(Y,X,D)
```

SS of DFS:

- For Directly connected cities.



- For indirectly connected cities.



Best First Search.

Best First search is an uninformed technique which uses heuristics to calculate the optimal path and for that our main goal is to find/generate all the heuristics h(n). As generating all the heuristics will be very tedious so I am fixing one destination which is Vishakhapatnam and calculating heuristics for that destination and saving it in a file named “heuristics.pl” I am generating this file using a python script.

The code for BFS is as follows :-

```
% best first search
bestFS([[Destination|Route]|_], Destination, [Destination|Route], 0).
bestFS([Route|Row],Destination,FinalRoute, NodeCount) :-
    find_successor(Route, NewRoutes),
    append(Row,NewRoutes,Row_One),
    sort_row(Row_One, NewRow),
    bestFS(NewRow, Destination, FinalRoute, Nodes),
    NodeCount is Nodes+1.


% sort the nodes w.r.t their costs
sort_row(L,L2) :-
    swap_nodes(L,L1), !,
    sort_row(L1,L2).
sort_row(L,L).


% swapping nodes according to the heuristic value.
swap_nodes([[A1|B1],[A2|B2]|T],[[A2|B2],[A1|B1]|T]) :-
    heuristic_distance(A1,W1),
    heuristic_distance(A2,W2),
```

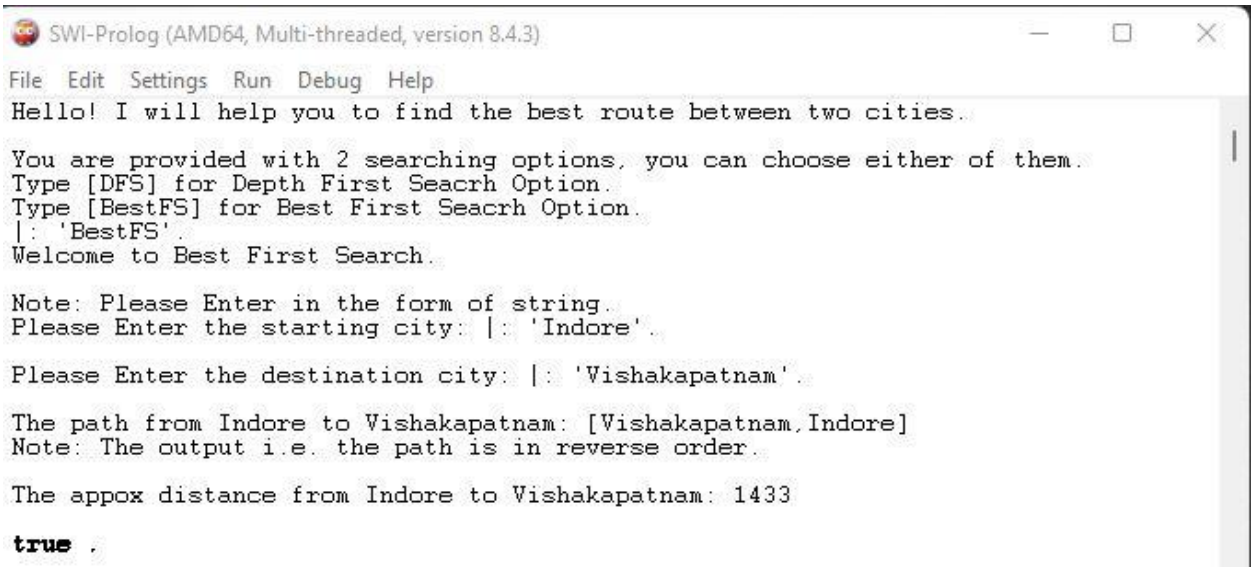
```
    W1>W2.
swap_nodes([X|T],[X|V]) :- swap_nodes(T,V).


% utility program to explore adjacent nodes
find_successor([Node|Route],NewRoute) :-
   findall([NewNode,Node|Route],
   (dist(Node, NewNode, _), \+ member(NewNode, Route)), NewRoute).

path_cost([Start,End],Cost) :- distance(Start,End,Cost).
path_cost([Start,End|Intermediate],Cost) :-
    distance(Start,End,Cost1),
    path_cost([End|Intermediate],Cost2),
    Cost is Cost1+Cost2.
```

SS of Best First Search:

- For Indirectly connected cities

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File  Edit  Settings  Run  Debug  Help
Hello! I will help you to find the best route between two cities.

You are provided with 2 searching options, you can choose either of them.
Type [DFS] for Depth First Seacrh Option.
Type [BestFS] for Best First Seacrh Option.
|: 'BestFS'.
Welcome to Best First Search.

Note: Please Enter in the form of string.
Please Enter the destination city: |: 'Hubli'.
Please enter the starting city: |: 'Vishakapatnam'.

The path is: [Vishakapatnam,Bhubaneshwar,Hubli]
The distance is: 2065
```

- For directly connected cities.


SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
Hello! I will help you to find the best route between two cities.
You are provided with 2 searching options, you can choose either of them.
Type [DFS] for Depth First Seacrh Option.
Type [BestFS] for Best First Seacrh Option.
|: 'BestFS'.
Welcome to Best First Search.
Note: Please Enter in the form of string.
Please Enter the starting city: |: 'Indore'.
Please Enter the destination city: |: 'Vishakapatnam'.
The path from Indore to Vishakapatnam: [Vishakapatnam,Indore]
Note: The output i.e. the path is in reverse order.
The appox distance from Indore to Vishakapatnam: 1433
true .