AI – Assignment 2

Write a Prolog program to search a road route from any city to any other city.

Assumption:

- All the roads bidirectional.

The XLS sheet is converted into csv file using an inbuilt function, then Depth-first search and Best-First Search is applied on the given knowledge base.

Source Code:

```
:- [library(csv)] .

:- [library(lists)] .

:- dynamic distance/3 .


:- op(1,'xfy','csv_') .


csv(FILE0)

:-

(start) csv_ (FILE0)

.


(start) csv_ (FILE0)

:-

csv:csv_read_file(FILE0,[HEADER|ROWss]) ,

row__to__list(HEADER,HEADERs) ,

(loop) csv_ (HEADERs,ROWss)

.


(loop) csv_ (_HEADERs,[])

:-

true

.
```

```
(loop) csv_ (HEADERs,[ROW|ROWss])
:-
row__to__list(ROW,ROWs) ,
lists:nth1(1,ROWs,CITY_A) ,
QUERY_A=(lists:nth1(NTH,ROWs,DISTANCE)) ,
QUERY_B=(NTH > 1) ,
QUERY_C=(lists:nth1(NTH,HEADERs,CITY_B)) ,
QUERY=(QUERY_A,QUERY_B,QUERY_C) ,
ASSERT=assertz(distance(CITY_A,CITY_B,DISTANCE)) ,
forall(QUERY,ASSERT) ,
(loop) csv_ (HEADERs,ROWss)
.

row__to__list(ROW,ROWs)
:-
ROW=..[_|ROWs]
.
dfs([[X|Y]|_],X,[X|Y]).
dfs([Y|Q],X,FP) :-
   add(Y,NP),
   append(NP,Q,NQ),
   dfs(NQ,X,FP).

bestfs([[X1|Y1]|_],X1,[X1|Y1]).
bestfs([Y1|Q],X1,FP1) :-
   add(Y1,NP1),
   append(Q,NP1,Q1),
   sorting(Q1,NQ1),
   bestfs(NQ1,X1,FP1).
```

```
add([N|Y],NP) :-
    findall([NN,N|Y],
          (distance(N,NN,_),
           \+ member(NN,Y)), NP).

sorting(L,L2) :-
    swapping(L,L1), !,
    sorting(L1,L2).
sorting(L,L).

swapping([[A1|B1],[A2|B2]|T],[[A2|B2],[A1|B1]|T]) :-
    hh(A1,W1),
    hh(A2,W2),
    W1>W2.
swapping([X|T],[X|V]) :-
    swapping(T,V).

hh(State, Value) :-
    h(State,Value),
    number(Value), !.
hh(State, Value) :-
   write('WrongFunction: '),
   write(h(State, Value)), nl,
   abort.

h(_,1).
```

Output:

```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File  Edit  Settings  Run  Debug  Help
%  library(csv) compiled into csv 0.02 sec, 0 clauses
%  library(lists) compiled into lists 0.00 sec, 0 clauses
% c:/Users/sehba/Documents/Prolog/readingcsv.pl compiled 0.05 sec, -4 clauses
?- csv('roaddistance.csv').
true .

?- dfs([['Hubli']],'Pune',DFSearchRoute).
DFSearchRoute = ['Pune', 'Pondicherry', 'Meerut', 'Patna', 'Madurai', 'Panjim', 'Ludhiana', 'Nasik', 'Kolhapur', 'Nagpur', 'Jullundur', 'Madras', 'Jamshedpur', 'Lucknow', 'Jabalpur', 'Kanpur', 'Imphal', 'Jaipur', 'Calicut', 'Indore', 'Gwalior', 'Hyderabad', 'Coimbatore', 'Delhi', 'Bhopal', 'Cochin', 'Baroda', 'Chandigarh', 'Asansol', 'Calcutta', 'Amritsar', 'Bombay', 'Allahabad', 'Bhubaneshwar', 'Agra', 'Bangalore', 'Agartala', 'Ahmedabad', 'Hubli'] .

?- dfs([['Ahmedabad']],'Agartala',DFSearchRoute).
DFSearchRoute = ['Agartala', 'Ahmedabad'] .

?- dfs([['Ahmedabad']],'Agra',DFSearchRoute).
DFSearchRoute = ['Agra', 'Bangalore', 'Agartala', 'Ahmedabad'] .

?- dfs([['Bangalore']],'Asansol',DFSearchRoute).
DFSearchRoute = ['Asansol', 'Calcutta', 'Amritsar', 'Bombay', 'Allahabad', 'Bhubaneshwar', 'Agra', 'Ahmedabad', 'Agartala', 'Bangalore'] .

?- dfs([['Vijayawada']],'Jaipur',DFSearchRoute).
DFSearchRoute = ['Jaipur', 'Hubli', 'Indore', 'Gwalior', 'Hyderabad', 'Coimbatore', 'Delhi', 'Bhopal', 'Cochin', 'Baroda', 'Chandigarh', 'Asansol', 'Calcutta', 'Amritsar', 'Bombay', 'Allahabad', 'Bhubaneshwar', 'Agra', 'Bangalore', 'Agartala', 'Ahmedabad', 'Vijayawada'] 
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File  Edit  Settings  Run  Debug  Help
%  library(csv) compiled into csv 0.03 sec, 0 clauses
%  library(lists) compiled into lists 0.00 sec, -65 clauses
% c:/Users/sehba/Documents/Prolog/readingcsv.pl compiled 0.05 sec, 0 clauses
?- csv('roaddistance.csv').
true .

?- bestfs([['Hubli']],'Agartala',BestSearchRoute).
BestSearchRoute = ['Agartala', 'Ahmedabad', 'Hubli'] .

?- bestfs([['Ahmedabad']],'Agartala',BestSearchRoute).
BestSearchRoute = ['Agartala', 'Ahmedabad'] .

?- bestfs([['Pune']],'Agartala',BestSearchRoute).
BestSearchRoute = ['Agartala', 'Pune'] .

?- bestfs([['Pune']],'Asansol',BestSearchRoute).
BestSearchRoute = ['Asansol', 'Pune'] .

?- bestfs([['Jaipur']],'Asansol',BestSearchRoute).
BestSearchRoute = ['Asansol', 'Jaipur'] .

?- bestfs([['Agra']],'Bhopal',BestSearchRoute).
BestSearchRoute = ['Bhopal', 'Ahmedabad', 'Agra'] .

?- 
```