**LAB-ASSIGNMENT-7**

**I19MA006**

Implement 8 Puzzle problem

```
s([A,B,C],X,l):-
    member(0,A)-> (A=[H1,H2,H3],
                   H1\==0->
                   ((H2==0->[0,H1,H3]=A1;
                     H3==0->[H1,0,H2]=A1),
                    [A1,B,C]=X));
    member(0,B)-> (B=[H1,H2,H3],
                   H1\==0->
                   ((H2==0->[0,H1,H3]=B1;
                     H3==0->[H1,0,H2]=B1),
                    [A,B1,C]=X));
    member(0,C)-> (C=[H1,H2,H3],
                   H1\==0->
                   ((H2==0->[0,H1,H3]=C1;
                     H3==0->[H1,0,H2]=C1),
                    [A,B,C1]=X)).

s([A,B,C],X,r):-
    member(0,A)-> (A=[H1,H2,H3],
                   H3\==0->
                   ((H1==0->[H2,0,H3]=A1;
                     H2==0->[H1,H3,0]=A1),
                    [A1,B,C]=X));
    member(0,B)-> (B=[H1,H2,H3],
                   H3\==0->
                   ((H1==0->[H2,0,H3]=B1;
                     H2==0->[H1,H3,0]=B1),
                    [A,B1,C]=X));
    member(0,C)-> (C=[H1,H2,H3],
                   H3\==0->
                   ((H1==0->[H2,0,H3]=C1;
                     H2==0->[H1,H3,0]=C1),
                    [A,B,C1]=X)).
```

```
s([A,B,C],X,u):-
    not(member(0,A)),
    member(0,B)-> (A=[H1,H2,H3],
                   B=[H4,H5,H6],
                   (H4==0->[0,H2,H3]=A1,[H1,H5,H6]=B1;
                    H5==0->[H1,0,H3]=A1,[H4,H2,H6]=B1;
                    H6==0->[H1,H2,0]=A1,[H4,H5,H3]=B1),
                   [A1,B1,C]=X);
    member(0,C)-> (B=[H1,H2,H3],
                   C=[H4,H5,H6],
                   (H4==0->[0,H2,H3]=B1,[H1,H5,H6]=C1;
                    H5==0->[H1,0,H3]=B1,[H4,H2,H6]=C1;
                    H6==0->[H1,H2,0]=B1,[H4,H5,H3]=C1),
                   [A,B1,C1]=X).

s([A,B,C],X,d):-
    not(member(0,C)),
    member(0,A)-> (B=[H1,H2,H3],
                   A=[H4,H5,H6],
                   (H4==0->[0,H2,H3]=B1,[H1,H5,H6]=A1;
                    H5==0->[H1,0,H3]=B1,[H4,H2,H6]=A1;
                    H6==0->[H1,H2,0]=B1,[H4,H5,H3]=A1),
                   [A1,B1,C]=X);
    member(0,B)-> (C=[H1,H2,H3],
                   B=[H4,H5,H6],
                   (H4==0->[0,H2,H3]=C1,[H1,H5,H6]=B1;
                    H5==0->[H1,0,H3]=C1,[H4,H2,H6]=B1;
                    H6==0->[H1,H2,0]=C1,[H4,H5,H3]=B1),
                   [A,B1,C1]=X).

next([H|T],H,T).

len([H|T],S):-
    T==[]->S is 1;
    T\==[]->len(T,S1),
            S is S1+1.
```

```
nextlevel(H,T,V,Q):-
    len(H, S),
    (   S>1->next(H,H1,Tail);
        S==1->H=[H1]),
    H1\==[],
    [A,B,C]=H1,
    findall(P,(s([A,B,C],P,_),not(member(P,V))),N),
    (   member(T,N)->N=Q;
        not(member(T,N))-> (S==2-> N=Q;
                            S>2-> nextlevel(Tail,T,[V|N],Q1),
                                  [V,N]=N2,
                                  [N2|Q1]=Q)).

bfs(S,T):-
    S=V,
    len(S,N),
    (   N==3->[S,[]]=S1;
        N\==3->S=S1),
    nextlevel(S1,T,V,V1),
    (   member(T,V1)-> ([A,B,C]=T,
                        write(A),nl,write(B),nl,write(C));
        not(member(T,V1))-> bfs(V1,T)).

dfs(T,T,[T|V]):-
    [A,B,C]=T,
     write(A),nl,write(B),nl,write(C),nl,nl.
dfs(S,T,V):-
    [A,B,C]=S,
    s([A,B,C],X,_),
    not(member(X,V)),
    write(A),nl,write(B),nl,write(C),nl,nl,
    dfs(X,T,[X|V]).
```

**OUTPUT**

```
?- bfs([[1,2,3],[4,0,5],[6,7,8]],[[1,2,3],[4,5,0],[6,7,8]]).
[1,2,3]
[4,5,0]
[6,7,8]
true.

?- bfs([[1,2,3],[4,0,5],[6,7,8]],[[1,2,0],[4,5,3],[6,7,8]]).
[1,2,0]
[4,5,3]
[6,7,8]
true.
```

**UCS**: Since the cost of each step is 1, so the algorithm for Uniform Cost Search and Breadth-First Search will be the same.

**Time-Complexity:**

Depth First Search: $O(b^M)$.

Breadth-First Search: $O(b^D)$.

Uniform Cost Search: $O(b^D)$.

where,

**b**: Number of Branches. (On average for 3 × 3 matrix it is 3).

**M**: Maximum depth of the tree.

**D**: Depth at which goal state lies.

If time and space complexity are not taken into consideration, then BFS is better as it will return the shortest path to our goal state which is not possible with DFS (As, DFS will return the 1st possible path to goal state rather than the shortest one).