

Name: Brijesh Ramesbhai Rohit

Admission number: U19CS009

### AI-ASSIGNMENT-08

Implement N queens problem using below algorithms in prolog.

Compare the complexity of both algorithms.

Which algorithm is best suited for implementing the N queens problem and why?

1. Breadth First Search

CODE :

```
nqueen:-
    write("Enter N: "),
    read(N),
    helper([], [], 0, 0, N, Added),
    write(Added), nl,
    solve(Added, N).

len([_|T], L):-
    len(T, L1),
    L is L1+1.

addEle(A, List, NewList):-
    append(List, [A], NewList).

solve([], _):-
    write("No solution exists"), nl.

solve([Board|Rem], N):-
    len(Board, L1), nl,
    L1 == N ->
        write("Answer: "),
        write(Board), nl;
    len(Board, L2),
    helper(Rem, Board, L2, 0, N, Added),
    write(Added), nl,
    solve(Added, N).

helper(Boards, _, _, N, N, Boards).

helper(Boards, Board, R, C, N, Added):-
```

```

    %write(R),nl,
    C < N,
    ok(R,C,Board,0) ->
    addEle(C,Board,NewBoard),
    addEle(NewBoard,Boards,NewBoards),
    C1 is C+1,
    helper(NewBoards,Board,R,C1,N,Added);
    C2 is C+1,

helper(Boards,Board,R,C2,N,Added).

ok(_,_,[],_).

ok(R,C,[H|T],N):-
    H =\= C,
    N =\= R,
    Sum1 is R + C,
    Sum2 is H + N,
    Sum1 =\= Sum2,
    Diff1 is R - C,
    Diff2 is N-H,
    Diff1 =\= Diff2,
    N1 is N+1,
ok(R,C,T,N1).

```

## OUTPUT :

```
?- nqueen.  
Enter N: 4.  
[[0],[1],[2],[3]]  
  
[[1],[2],[3],[0,2],[0,3]]  
  
[[2],[3],[0,2],[0,3],[1,3]]  
  
[[3],[0,2],[0,3],[1,3],[2,0]]  
  
[[0,2],[0,3],[1,3],[2,0],[3,0],[3,1]]  
  
[[0,3],[1,3],[2,0],[3,0],[3,1]]  
  
[[1,3],[2,0],[3,0],[3,1],[0,3,1]]  
  
[[2,0],[3,0],[3,1],[0,3,1],[1,3,0]]  
  
[[3,0],[3,1],[0,3,1],[1,3,0],[2,0,3]]  
  
[[3,1],[0,3,1],[1,3,0],[2,0,3],[3,0,2]]  
  
[[0,3,1],[1,3,0],[2,0,3],[3,0,2]]  
  
[[1,3,0],[2,0,3],[3,0,2]]  
  
[[2,0,3],[3,0,2],[1,3,0,2]]  
  
[[3,0,2],[1,3,0,2],[2,0,3,1]]  
  
[[1,3,0,2],[2,0,3,1]]  
  
Answer: [1,3,0,2]  
true .
```

## 2. Depth First Search

```
addEle(A,List,NewList):-
    append(List,[A],NewList).
solve(Board,N,_,Board,N).
solve(Board,R,C,Ans,N):-
    ok(R,C,Board,0),
    addEle(C,Board,NewBoard),
    R1 is R+1,
    solve(NewBoard,R1,0,Ans,N).
solve(Board,R,C,Ans,N):-
    C < N-1,
    C1 is C+1,
    solve(Board,R,C1,Ans,N).
    ok(_,_,[],_).
ok(R,C,[H|T],N):-
    H =\= C,
    N =\= R,
    Sum1 is R + C,
    Sum2 is H + N,
    Sum1 =\= Sum2,
    Diff1 is R - C,
    Diff2 is N-H,
    Diff1 =\= Diff2,
    N1 is N+1,
    ok(R,C,T,N1).
:-
    write("Enter N: "),
    read(N),
    solve([],0,0,Ans,N),
    write(Ans).
```

OUTPUT :

```
?- consult('nqueen-dfs.pl').
Enter N: 8.
[0,4,7,5,2,6,1,3]
true.
```