

AIML Lab 4

U18CO021: SAHIL BONDRE

Question:

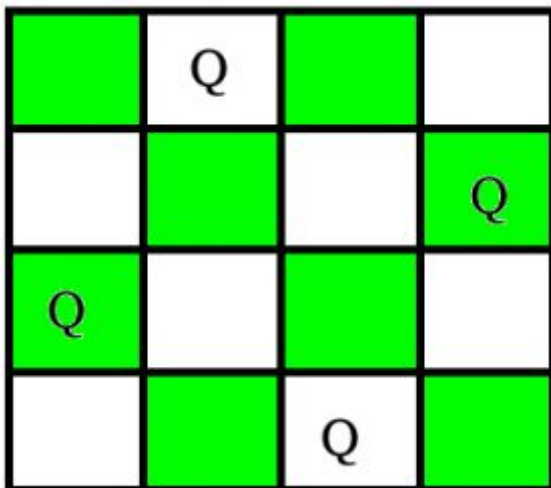
Implement N queens problem using below algorithms in prolog. Compare the complexity of both algorithms. Which algorithm is best suited for implementing N queens problem and why ?

1. Breadth First Search

2. Depth First Search

Solution:

The N-Queens problem is a classic problem that is often used in discussions of various search strategies. The problem is often defined in terms of a standard 8-by-8 chess board, although it can be defined for any N-by-N board.



This problem can be solved using various techniques like BFS, DFS, Backtracking etc.

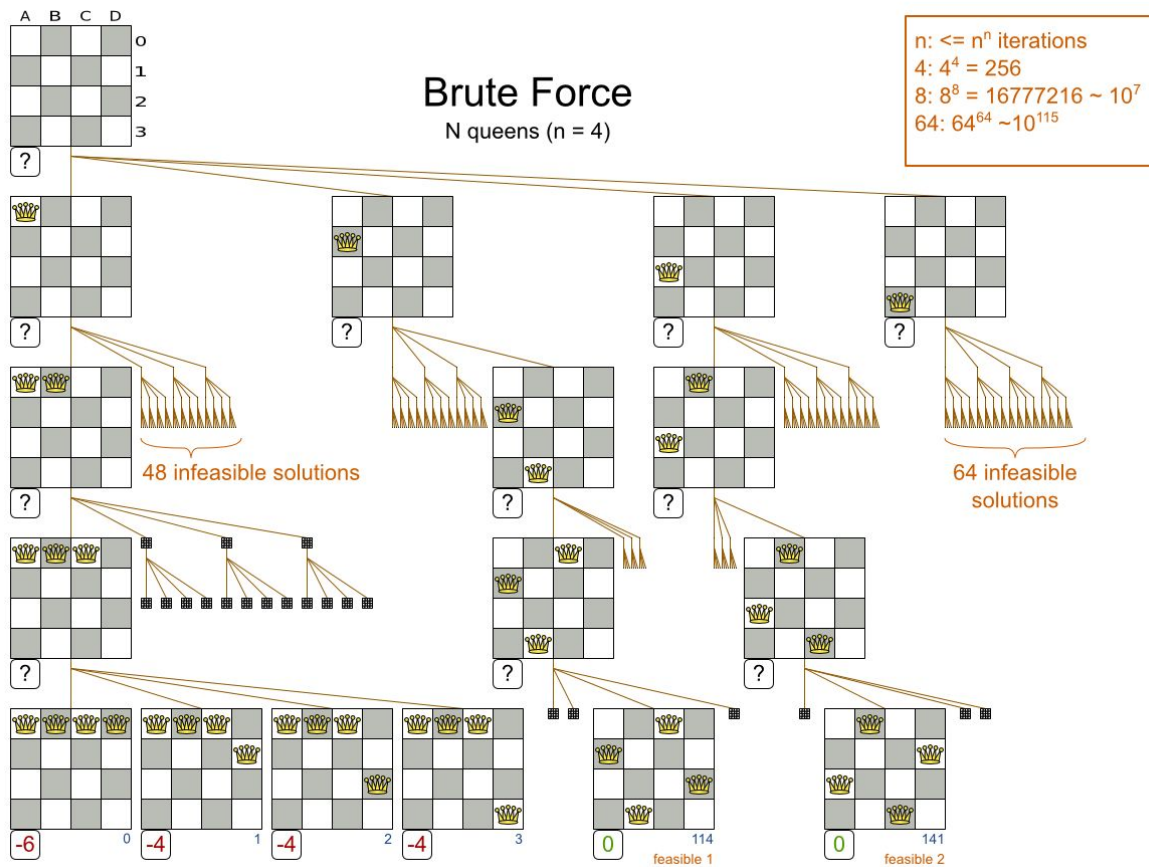
BFS takes $O(n^n)$ time to solve as it tries every possible solution

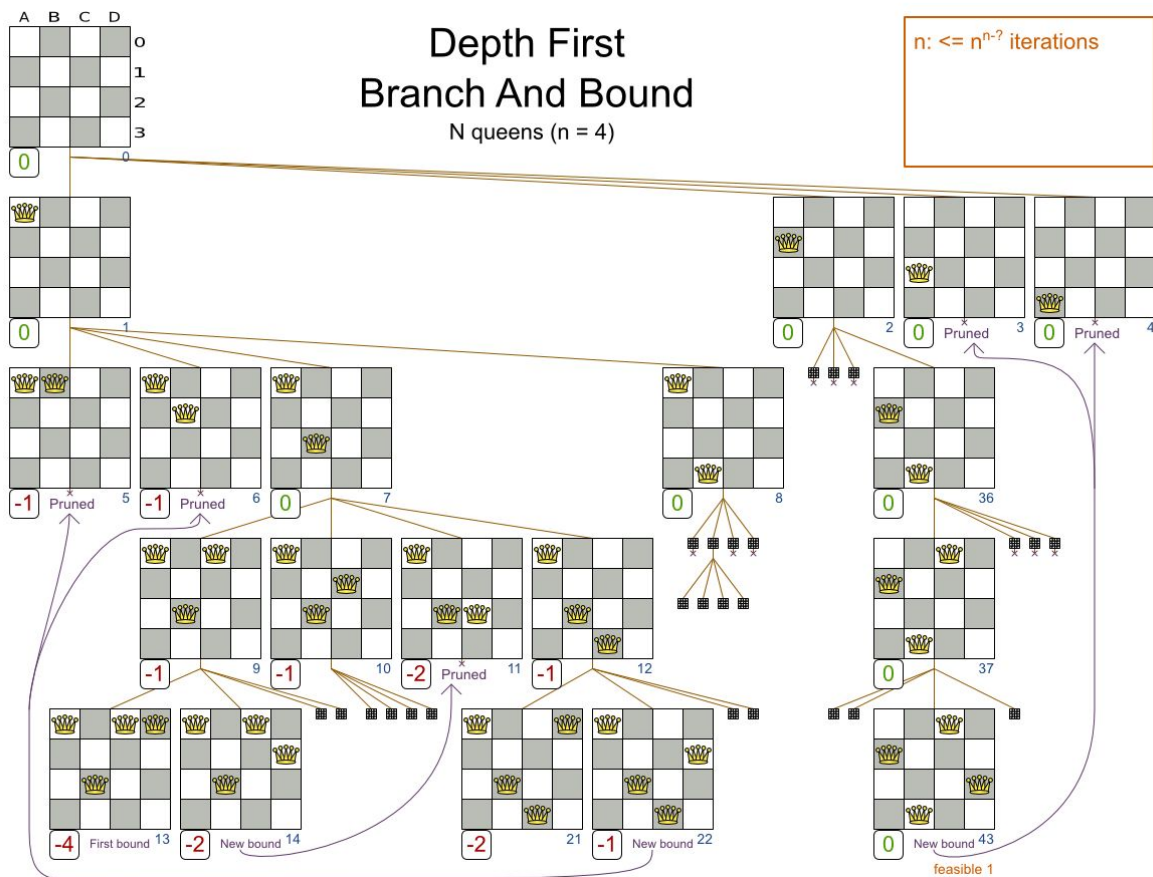
DFS takes $O(n!)$ time as it discards the invalid solutions and their following recursive calls as and when they are found

Backtracking solves this problem in $O(n!)$ time

Illustrations of BFS and DFS approach:

(<https://stackoverflow.com/questions/52089745/can-we-solve-4-queen-problem-using-best-first-search>)





BFS:

```

bfs(Ylist) :-
solve(Ylist,[1,2,3,4,5,6,7,8],
  [1,2,3,4,5,6,7,8],
  [-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7],
  [2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]).
solve([],[],[],Du,Dv).
solve([Y|Ylist],[X|Dx1],Dy,Du,Dv) :-
delete(Y,Dy,Dy1),
U is X-Y,
delete(U,Du,Du1),
V is X+Y,
delete(V,Dv,Dv1),
solve(Ylist,Dx1,Dy1,Du1,Dv1).
delete(Item,[Item|List],List).
delete(Item,[First|List],[First|List1]) :-
delete(Item,List,List1).
  
```

```
?- findall(S, bfs(S), LS), length(LS, N), write(N).
92
LS = [[1, 5, 8, 6, 3, 7, 2, 4], [1, 6, 8, 3, 7, 4, 2|...], [1, 7, 4, 6, 8, 2|...], [1, 7, 5, 8, 2|...], [2, 4, 6, 8|...], [2, 5, 7|...], [2, 5|...], [2|...], [...|...]|...],
N = 92.
```

DFS:

```
dfs(Queens) :-
permutation([1,2,3,4,5,6,7,8], Queens),
safe(Queens).
permutation([], []).
permutation([Head|Tail], PermList) :-
permutation(Tail, PermTail),
del(Head, PermList, PermTail).
del(Item, [Item|List], List).
del(Item, [First|List], [First|List1]) :-
del(Item, List, List1).
safe([]).
safe([Queen|Others]) :-
safe(Others),
noattack(Queen, Others, 1).
noattack(_, [], _).
noattack(Y, [Y1|Ylist], Xdist) :-
Y1-Y=\=Xdist,
Y-Y1=\=Xdist,
Dist1 is Xdist + 1,
noattack(Y, Ylist, Dist1).
```

```
?- findall(S, dfs(S), LS), length(LS, N), write(N).
92
LS = [[5, 2, 6, 1, 7, 4, 8, 3], [6, 3, 5, 7, 1, 4, 2|...], [6, 4, 7, 1, 3, 5|...], [3, 6, 2, 7, 5|...], [6, 3, 1, 7|...], [6, 2, 7|...], [6, 4|...], [3|...], [...|...]|...],
N = 92.
```