Modern Education Society's

Wadia College of Engineering, Pune

NAME OF STUDENT:	CLASS:
SEMESTER/YEAR:	ROLL NO:
DATE OF PERFORMANCE:	DATE OF SUBMISSION:
EXAMINED BY:	EXPERIMENT NO: LP-III(DAA)-05

TITLE: N Queen's Problem using Backtracking.

AIM: Design n-Queens matrix having first Queen placed. Use backtracking to place remaining Queens to generate the final n-queen's matrix

OBJECTIVES:

- To understand the concept of Backtracking.
- To understand the concept of State Space Tree.
- To compare the space & time complexity of Recursive & Non-Recursive techniques of Backtracking.

PRE-REQUISITES: Backtracking formulation is used to solve problems which deal with searching for a set of solutions or which ask for an optimal solution satisfying some constraints.

Definition of state space tree:-

The tree organization of the solution space is referred to as state space tree.

If the tree organization is independent of the problem instance being solved, they are called static trees.

If the tree organization is dependent of the problem instance being solved, they are called dynamic trees.

THEORY:

What is Backtracking?

Backtracking formulation is used to solve problems which deal with searching for a set of solutions or which ask for an optimal solution satisfying some constraints.

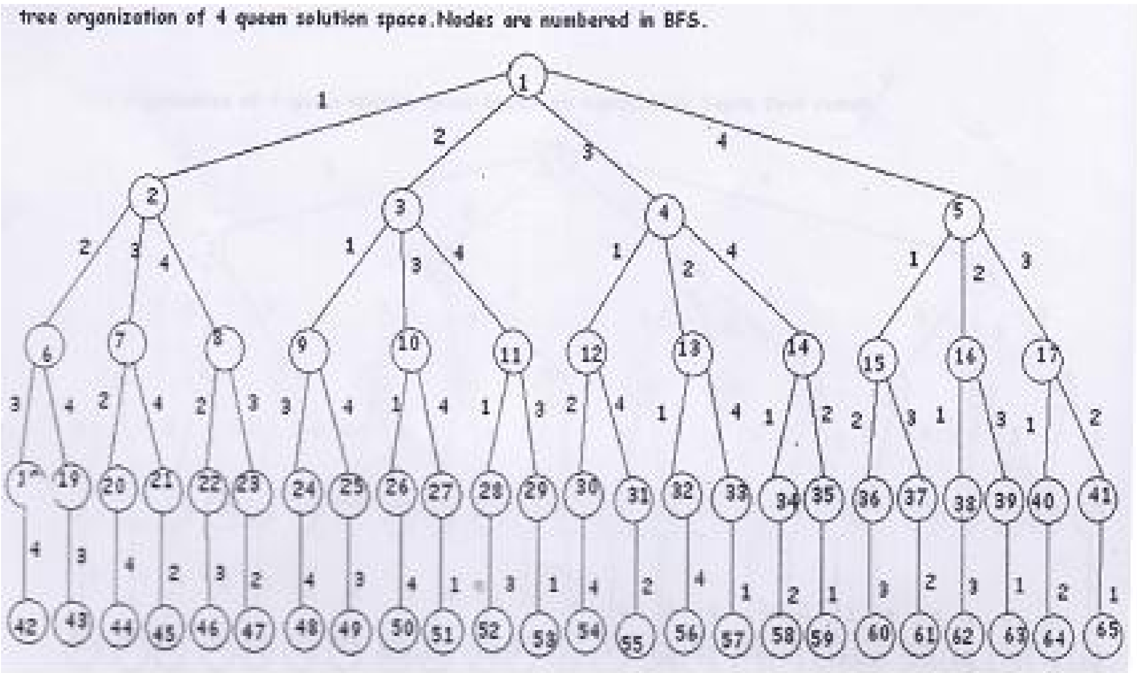
Constraint satisfaction problems:-

1. These are problems with complete solution, where the order of elements does not matter.
2. The problem consists of set of variables each of which must be assigned a value, subject to particular constraints of problem.
3. Backtracking attempts to try all the combinations in order to obtain a solution.
4. Its strength is that many implementations avoid trying many partial combinations, thus speeding up the running time.

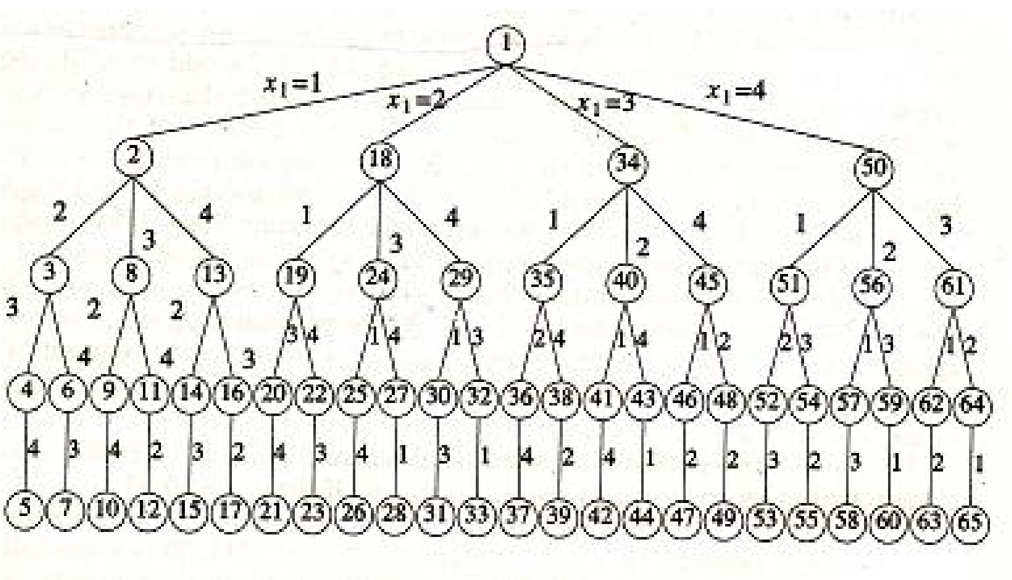
State space trees:-Backtracking algorithms determined problems solution by systematically searching the solution space for the given problem instance.This search is facilitated by using a tree organization for the solution space. For a given problem space many tree organizations are possible.

Consider a 4 * 4 chessboard. We have to place 4 queens such that no two queens are on the same row, column or diagonal. For the solution to this problem ,we place each of the 4 queens on a separate row. Now we have to place each queen on a unique column too such that no 2 queens are on same diagonals. Various permutations of queens position are possible but only the permutations that satisfy the constraints are valid. The Various permutations of queens positions can be depicted by a tree organization .Let the level of the tree denote the row and edge denote the column. Nodes denote the states reached. Definition of state space tree:-

The tree organization of the solution space is referred to as state space tree. If the tree organizations are independent of the problem instance being solved, they are called static trees. If the tree organization is dependent of the problem instance being solved, they are called dynamic trees.



tree organization of 4 queen space. Nodes are numbered in DFS



Implementation of backtracking algorithm for 4 queen's problem:-

Step1: We place the first queen in row 1, column 1 as shown.

1			

Step 2: Now we have to place queen 2 in row 2.We choose column 3 because no queens should be in same column or diagonal.

1			
		2	

Step 3: Now we have to place queen 3 in row 3.We can’t choose column 2, 1 or 3. So we backtrack and shift the queen to column 4. Then we place queen 3 in row 3, column 2 as shown.

1			
		2	

Step 4: now we have to place queen 4 in row 4. We cannot place queen 4 in column 1, 2, 3 or 4 since no queen should be in same column or diagonal.

1			
			2
	3		

Step 5: Thus we backtrack. We cannot shift queen 3 and still satisfy implicit constraints. We cannot move queen 2 and still satisfy constraints. Thus we shift queen 1 to column 2, row 1.

1			
			2
	3		

Step 6: Now we place queen 2 in column 4, row 2.

	1		

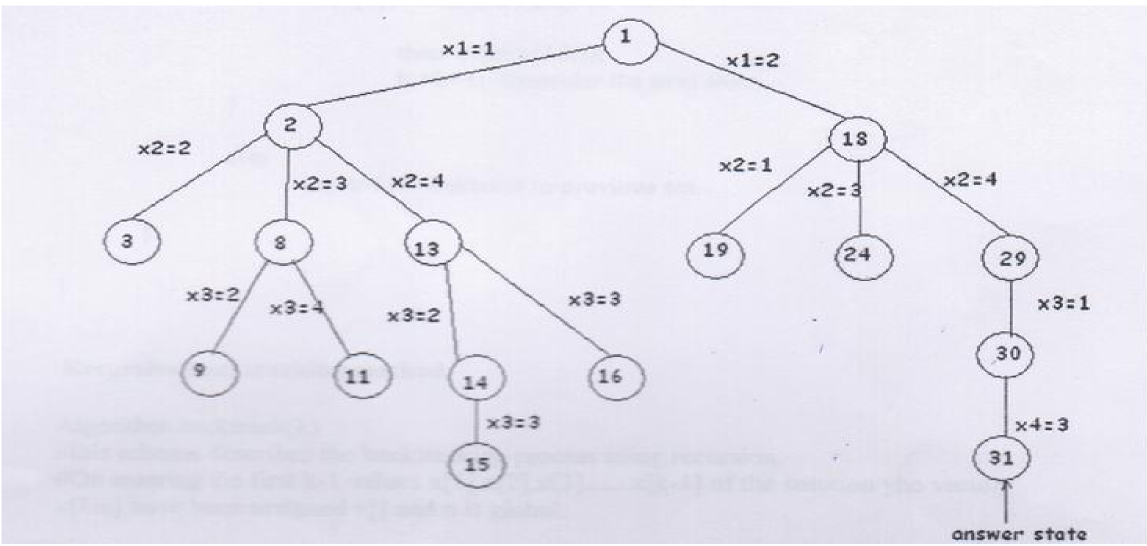
Step 7: Now we place queen 3 in column 1, row 3.

	1		
			2

Step 8: At last, we place queen in column 3, row 4.

	1		
			2
3			
		4	

The tree generated during the backtracking is as follows. The edges denote the column. The level(1) denotes the row of the queen(1).



ALGORITHM:-

Iterative backtracking Algorithm:

```
Algorithm Backtrack (n)
//This schema describes the backtracking process.
// all solutions are generated in x[1:n] and printed as soon as they are determined.
{
        k:= 1;
        While(k!=0)do
        {

                        if(there remains an untried x[k] belongs t(x[1],x[2],……, x[k- 1]) and
                         Bk(x[1],…...x[k] )is true) then
                        {
                                if(x[1],….x[k] is a path to answer node)
                                then write(x[1:k]);
                                k:=k.+1; //consider the next state.
                         }

                        else
                        k:=k-1; //backtrack to previous set.
        }
 }
```

Recursive backtracking method.

```
Algorithm backtrack(k)
//this schema describes the backtracking process using recursion.
//On entering the first k-1 values x[1],x[2],x[3],......x[k-1] of the solution the vector x[1:n] have
been assigned x[] and n is global.
{For (each x[k]belongs t(x[1],x[k-1]))
do{if Bk(x [1], x[2],....x[k]!=0)then{
IfBk (x[1],x[2],……x[k] is a path to answer node)
then wrie (x[1:k]);
if(k<n) then backtrack(k+1);
}
}
}
```

CONCLUSION: Implemented 8 queen successfully.

QUESTIONS FOR REVIEW:

1) What is backtracking? What are the peculiar characteristics & applications of this approach?
2) Explain Explicit & Implicit constraints with respect to 8 queen's problem?
3) Compare the space & time complexity of Recursive & Non-Recursive techniques of Backtracking.
4) Write realistic applications of this experiment in brief (at least two applications).