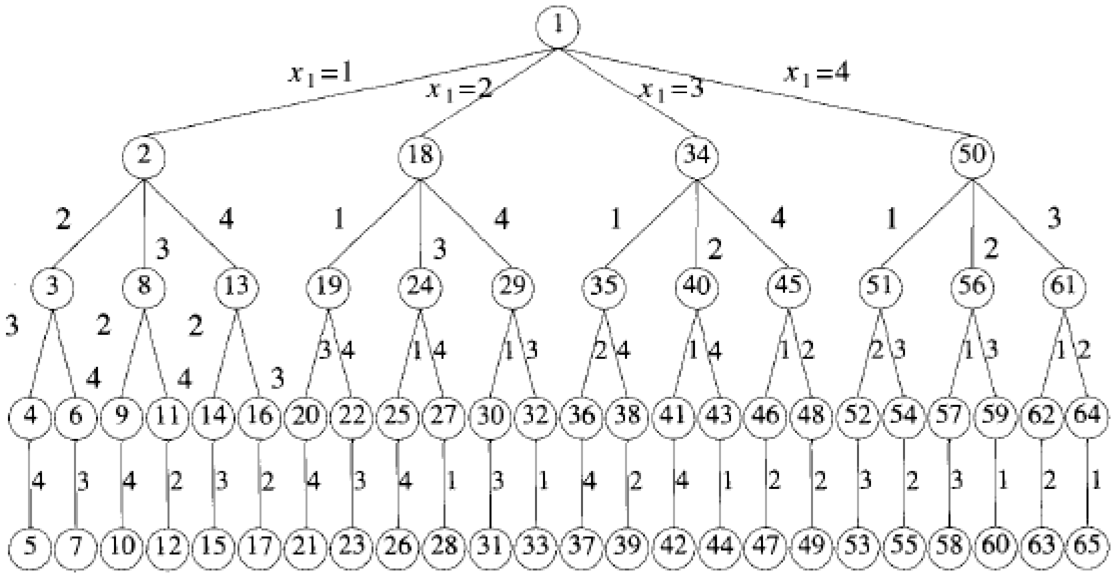
Experiment No. 13

Title: Write a program to find solution to N Queen Problem using backtracking

Theory:

The n-queen's problem is a generalization of the 8-queens or 4-queens problem. Now n queens are to be placed on an n x n chess board so that no two attack; that is, no two queens are on the same row, column, or diagonal. The solution space consists of all n! permutations of the n-tuple(1,2,,n.). Figure shows a possible tree organization for the case n = 4. A tree such as this is called a permutation tree. The edges are labeled by possible values of Xi. Edges from level 1 to level 2 nodes specify the values for x1. Thus, the leftmost subtree contains all solutions with x1 = 1; its leftmost subtree contains all Solutions with x1 = 1 and x2 = 2, and soon. Edges from level i to level i +1 Are labeled with the values of xi. The solution space is defined by all paths from the root node to a leaf node. There are 4!= 24 leaf nodes in the tree

of Figure



The condition to check that no two queens are placed in attacking position diagonally can be formulated as

Suppose two queens are placed at positions (i,j) and (k,l). then by above they are on the same diagonal only if

i-j=k-l or i+j =k+l

The first equation implies

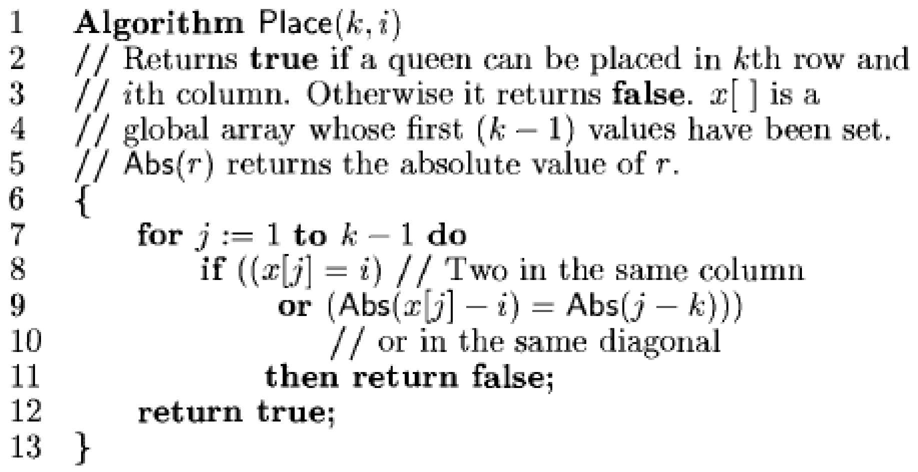
j-l=i-k

the second condition implies

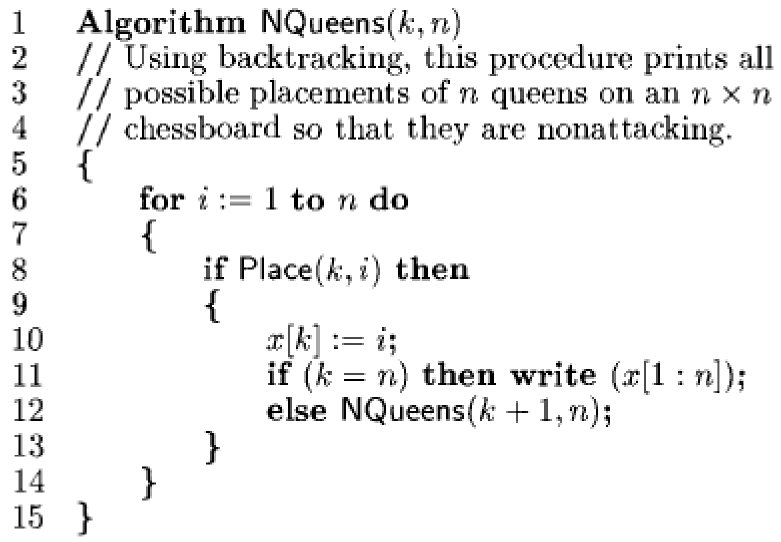
j-l=k-i

therefore two queens lie on the same diagonal if and only if |j-l| = |i-k|

Algorithm Place(k,i) returns true if Kth queen can be placed in column i. it tests both whether I is distinct from all previous x[1],x[2] …x[k-1] and whether there is no other queen on the same diagonal. Its computing time is O(k-1). The arra x[] is global. The algorithm is invoked by NQeens(1, n)



```
Algorithm Place($k, i$)
// Returns true if a queen can be placed in $k$th row and
// $i$th column. Otherwise it returns false. $x[\,]$ is a
// global array whose first $(k-1)$ values have been set.
// Abs($r$) returns the absolute value of $r$.
{
    for $j := 1$ to $k-1$ do
        if (($x[j] = i$) // Two in the same column
            or (Abs($x[j] - i$) = Abs($j - k$)))
                // or in the same diagonal
            then return false;
    return true;
}
```



```
Algorithm NQueens($k, n$)
// Using backtracking, this procedure prints all
// possible placements of $n$ queens on an $n \times n$
// chessboard so that they are nonattacking.
{
    for $i := 1$ to $n$ do
    {
        if Place($k, i$) then
        {
            $x[k] := i$;
            if ($k = n$) then write ($x[1:n]$);
            else NQueens($k+1, n$);
        }
    }
}
```

Complexity:

For an 8 x 8 chessboard there are ${}^{64}C_8$ possible ways to place 8 pieces, or approximately 4.4 billion 8-tuples to examine. However, by Allowing only placements of queens on distinct rows and columns, we require the examination of at most 8!,or only 40,320 8-tuples hence complexity is O(n!)