

N	Solutions	Recursive calls
8	92	2056
10	724	35,538
12	14,200	856,188

It's pretty clear that with little increase to N , the solutions and recursive calls increase exponentially. When N was equal to 12, it took about 22 seconds on average to finish execution, whereas when N equals 10, it takes on average 0.7 seconds and 0.03 seconds when $N=8$.

Process (A , k , S) :

if length of S > count

count += 1

return

Is Solution (A , k , S) :

if $k \geq \text{length of } A$

check queen count is equal to n in A

if queen count = n

S. Append (A)

return True

return False

Construct Candidates (A , row , col) :

check queen position threat

at A[row][col] for row and

col positions as candidates

if A[row][col] = safe

return True

else

return False

Backtrack (A, k, S):

if IsSolution (A, k, S):

Process (A, k, S)

else

for row in A

if ConstructCandidates (A, row, k, S):

$A[i][k] = 1$

Backtrack ($A, k+1, S$)

if Finished():

return

$A[i][k] = 0$

Finished()

return False

The time complexity of the Backtracking algorithm to solve the N-Queen Problem is $O(n!)$ where n is the number of queens.