Backtracking

Srivaths P

Recursive Call Stack

When a recursion is executed, the function calls are placed in a stack.

The first call is at the bottom of the stack, and the last call is at the top.

After a call is executed, it is popped from the stack, and the returned value is passed to the current top element.

Space complexity can be found using the size of the stack.

Backtracking

Backtracking is brute-force recursion, but it is far more efficient as it can skip over many unnecessary recursive calls.

Backtracking will stop at the first instance where some given condition is not fulfilled. This process will remove entire subtrees from the tree.

The time complexity is usually similar to brute-force, but it is much faster when applied.

Backtracking Example Problem

Given N positive coins, find number of combinations you can make such that the sum does not exceed K.

Brute-force will go till the last possible index, and then check if the sum does not exceed K.

Backtracking will stop at the first index where the sum exceeds K.

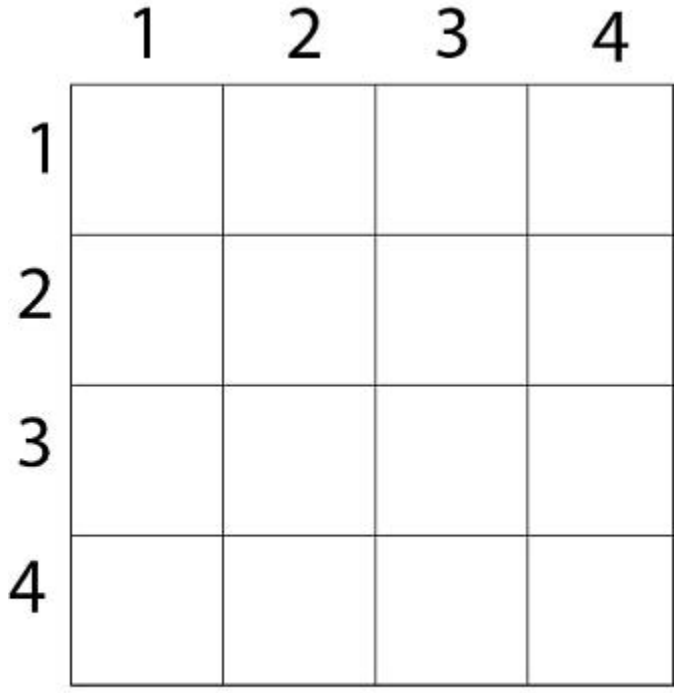
Sudoku

Fill a 9x9 grid with 1 to 9 such that such that each row, column, and 3x3 sub-grids have all values from 1 to 9.

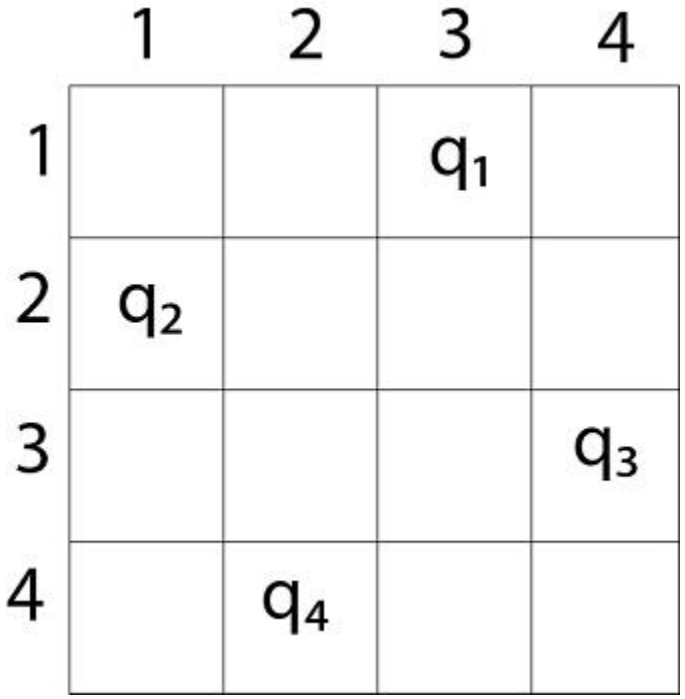
5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

N-Queens Problem

Given an NxN chessboard, find the number of arrangements of queens such that none of the queens attack each other.



4x4 chessboard



Problem Solving:

- https://codingcompetitions.withgoogle.com/kickstart/round/00000000008cb33e/00000000009e762e

- https://leetcode.com/problems/sudoku-solver/