Backtracking

The **backtracking algorithm** is a problem-solving approach that tries out all the possible solutions and chooses the best or desired ones. Generally, it is used to solve problems that have multiple solutions. The term backtracking suggests that for a given problem, if the current solution is not suitable, eliminate it and then backtrack to try other solutions.

Types of Backtracking Problem

The backtracking algorithm is applied to some specific types of problems. They are as follows −

- **Decision problem** − It is used to find a feasible solution of the problem.
- **Optimization problem** − It is used to find the best solution that can be applied.
- **Enumeration problem**− It is used to find the set of all feasible solutions of the problem.

Examples of Backtracking Algorithm

The following list shows some examples of Backtracking Algorithm −

- Hamiltonian Cycle
- M-Coloring Problem
- N Queen Problem
- Sum of subset

N Queen Problem

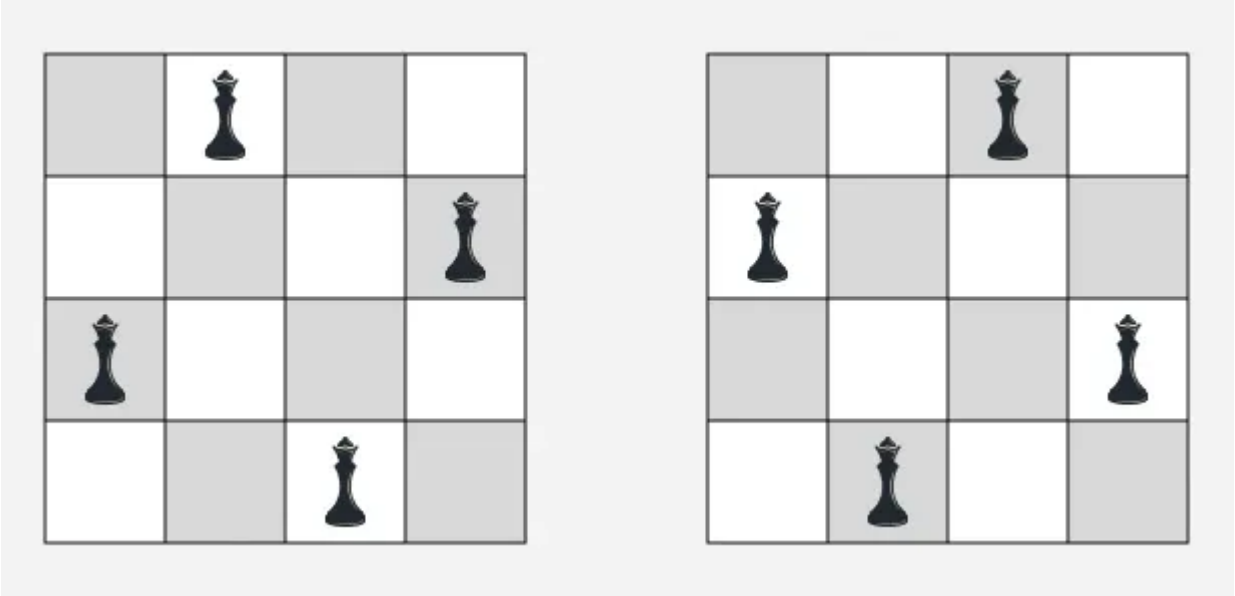
In N-Queen problem, we are given an NxN chessboard and we have to place N number of queens on the board in such a way that no two queens attack each other. A queen will attack another queen if it is placed in horizontal, vertical or diagonal points in its way. The most popular approach for solving the N Queen puzzle is Backtracking.

4 Queen Problem

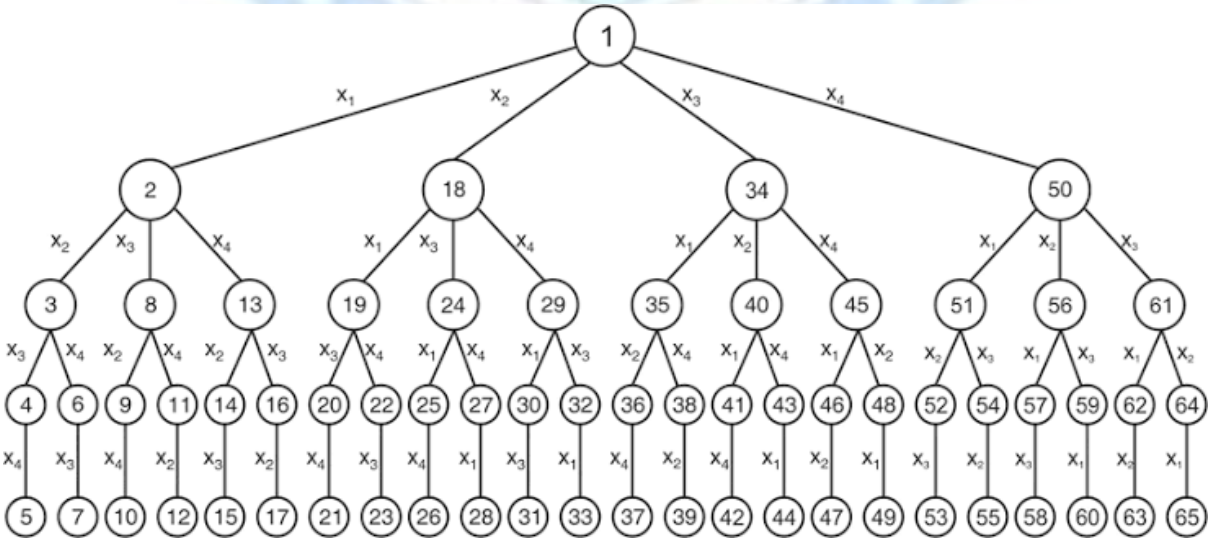
Input: n = 4

Output: [[2, 4, 1, 3], [3, 1, 4, 2]]

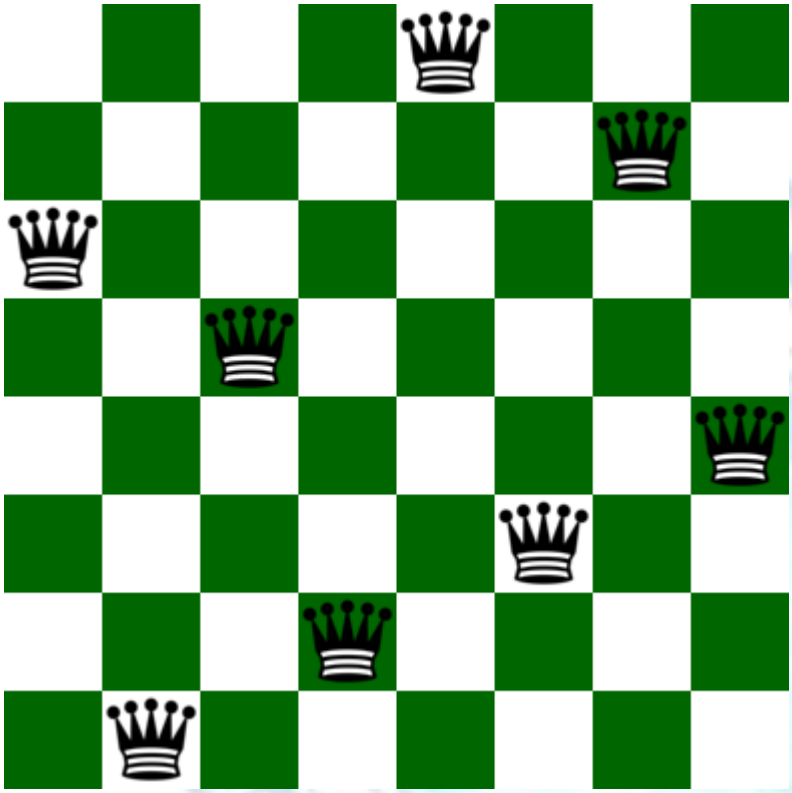
Explanation: We mainly print column numbers (from first to last row) of every possible configuration.



State Space Tree:



8 queen problem



Sum of Subsets:

In the **sum of subsets problem**, there is a given set with some non-negative integer elements. And another sum value is also provided, our task is to find all possible subsets of the given set whose sum is the same as the given sum value.

Set: In mathematical terms, a **set** is defined as a collection of similar types of objects. The entities or objects of a set must be related to each other through the same rule.

Subset: Suppose there are two sets namely set P and set Q. The set P is said to be a subset of set Q, only if all the elements of set P also belong to the set Q and vice-versa need not be true.

Suppose the given set and sum value is −

Set = {1, 9, 7, 5, 18, 12, 20, 15}

sum value = 35

All possible subsets of the given set, where sum of each element for every subset is the same as the given sum value are given below −

{1 9 7 18}

{1 9 5 20}

{5 18 12}

Backtracking Approach to solve Subset Sum Problem

In the naive method to solve a subset sum problem, the algorithm generates all the possible permutations and then checks for a valid solution one by one. Whenever a solution satisfies the constraints, mark it as a part of the solution.

In solving the subset sum problem, the backtracking approach is used for selecting a valid subset. When an item is not valid, we will backtrack to get the previous subset and add another element to get the solution.

In the worst-case scenario, backtracking approach may generate all combinations, however, in general, it performs better than the naive approach.

Follow the below steps to solve subset sum problem using the backtracking approach

- First, take an empty subset.
- Include the next element, which is at index 0 to the empty set.
- If the subset is equal to the sum value, mark it as a part of the solution.
- If the subset is not a solution and it is less than the sum value, add next element to the subset until a valid solution is found.
- Now, move to the next element in the set and check for another solution until all combinations have been tried.