## UNIT-5

# BACKTRACKING

**Prof. Ritesh Upadhyay**

Faculty of Engineering and Te

Ganpat University

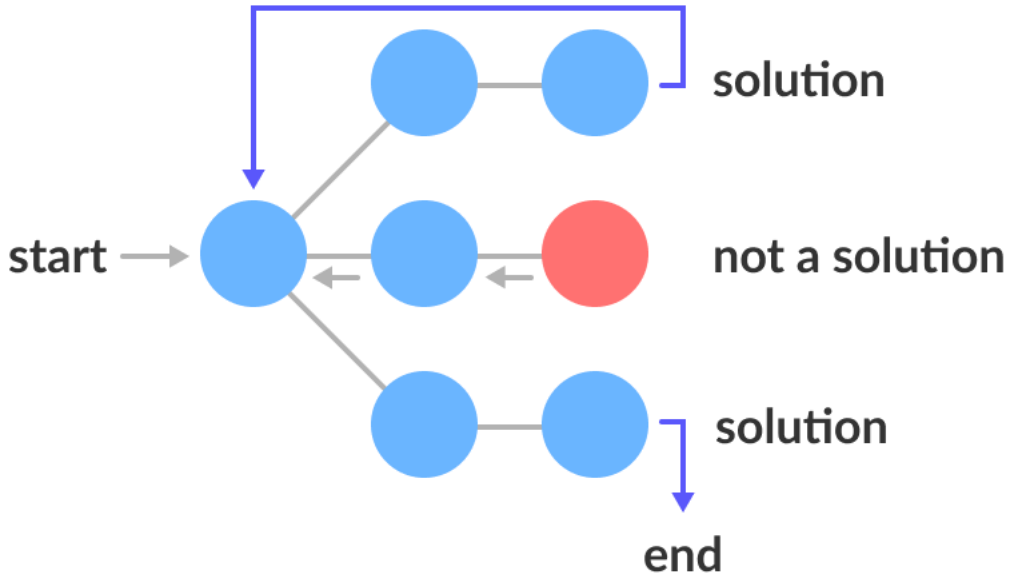
॥ विद्यया समाजोत्कर्षः ॥

# Backtracking Algorithm

- A backtracking algorithm is a problem-solving algorithm that uses a brute force approach for finding the desired output.
- The Brute force approach tries out all the possible solutions and chooses the desired/best solutions.
- The term backtracking suggests that if the current solution is not suitable, then backtrack and try other solutions. Thus, recursion is used in this approach.
- Backtracking algorithms use Depth first search (DFS) for finding the solution of the problem.
- Backtracking is an algorithmic-technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time.
- Backtracking can be defined as a general algorithmic technique that

# State Space Tree

A space state tree is a tree representing all the possible states (solution or non-solution) of the problem from the root as an initial state to the leaf as a terminal state.

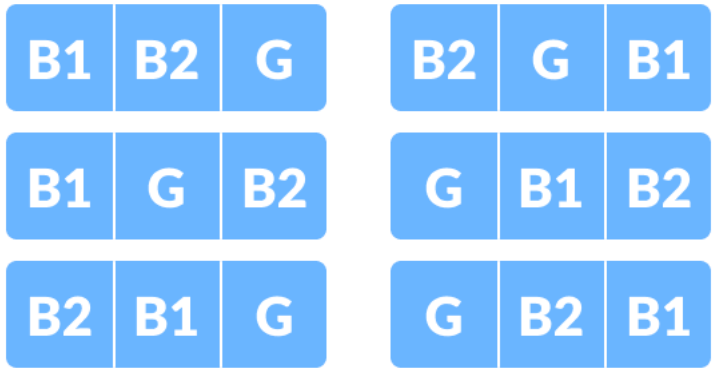


# Example Backtracking Approach

**Problem**: You want to find all the possible ways of arranging 2 boys and 1 girl on 3 benches. Constraint: Girl should not be on the middle bench.

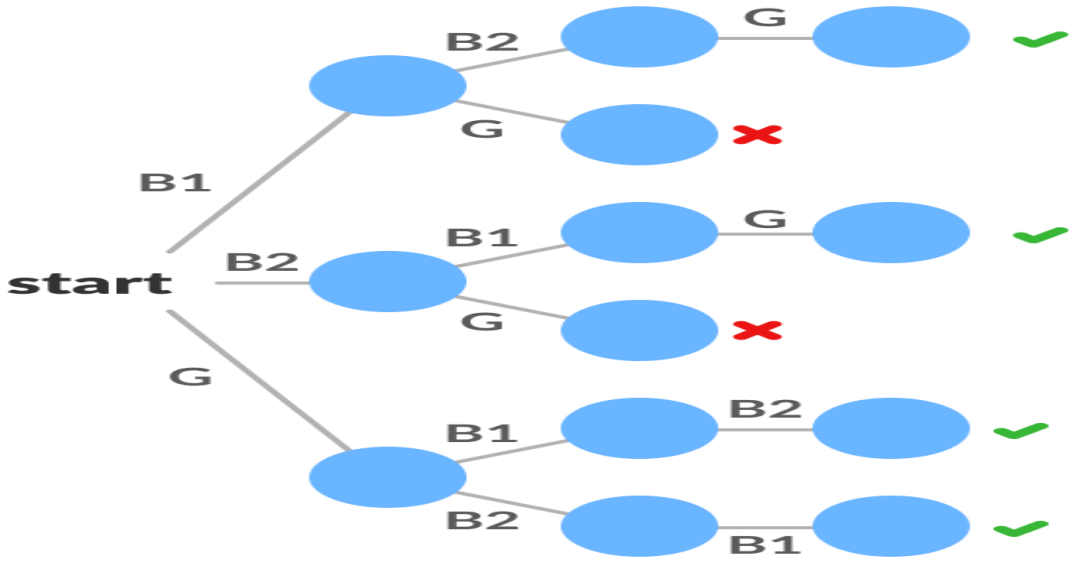
**Solution:** There are a total of 3! = 6 possibilities. We will try all the possibilities and get the possible solutions. We recursively try all the possibilities.

All the possibilities are:



# Example Backtracking Approach

- The following state space tree shows the possible solutions



# Backtracking Algorithm Applications

- Maze solving problem.
- The Knight's tour problem.
- Sudoku
- N-Queens Problem
- The Knapsack Problem
- Generalized Strings
- Hamiltonian Cycles
- Graph Coloring Problem

# Advantages of Backtracking Approach

- comparison with the Dynamic Programming, Backtracking Approach is more effective in some cases
- Backtracking Algorithm is the best option for solving tactical problem.
- Also Backtracking is effective for constraint satisfaction problem.
- In greedy Algorithm, getting the Global Optimal Solution is a long procedure and depends on user statements but in Backtracking It Can Easily getable.
- Backtracking technique is simple to implement and easy to code.
- Different states are stored into stack so that the data or Info can be usable anytime

# Disadvantages of Backtracking Approach

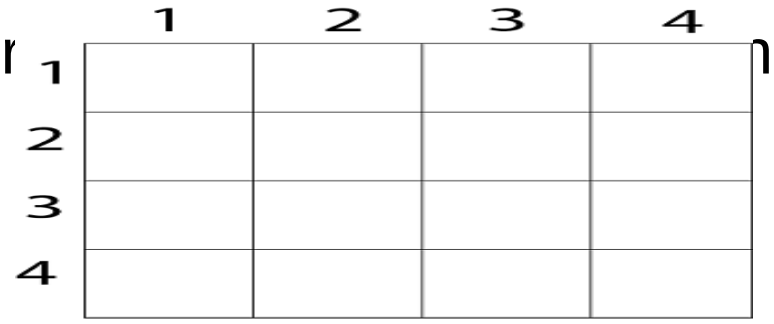
- Backtracking Approach is not efficient for solving strategic Problem
- The overall runtime of Backtracking Algorithm is normally slow
- To solve Large Problem Sometime it needs to take the help of other techniques like Branch and bound.
- Need Large amount of memory space for storing different state function in the stack for big problem.
- Thrashing is one of the main problem of Backtracking
- The Basic Approach Detects the conflicts too late.

# N-Queens Problem

- Problem Description In a N*N square board
- N – number of queens need to be placed considering three Condition ---

1. No two Queens can be placed in same row.
2. No two Queens Can be places in same Column
3. No two queens Can be placed in same Diagonal..

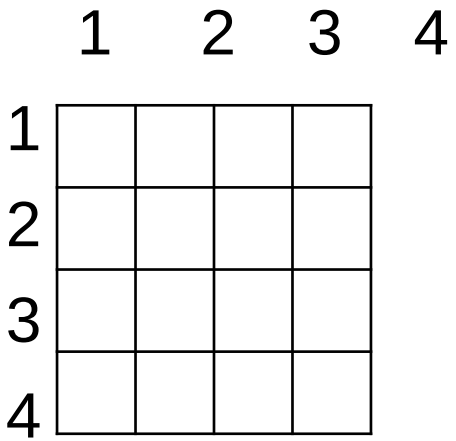
- It can be seen that for n =1, the problem has a trivial solution, and no solution exists for n =2 and n =3. So first we will consider the 4 queens problem and then generate it to n - queens problem.
- Given a 4 x 4 chessboard and number[illegible]ne chessboard 1 through 4
- Lets queen names are R,S T,U .



4x4 chessboard

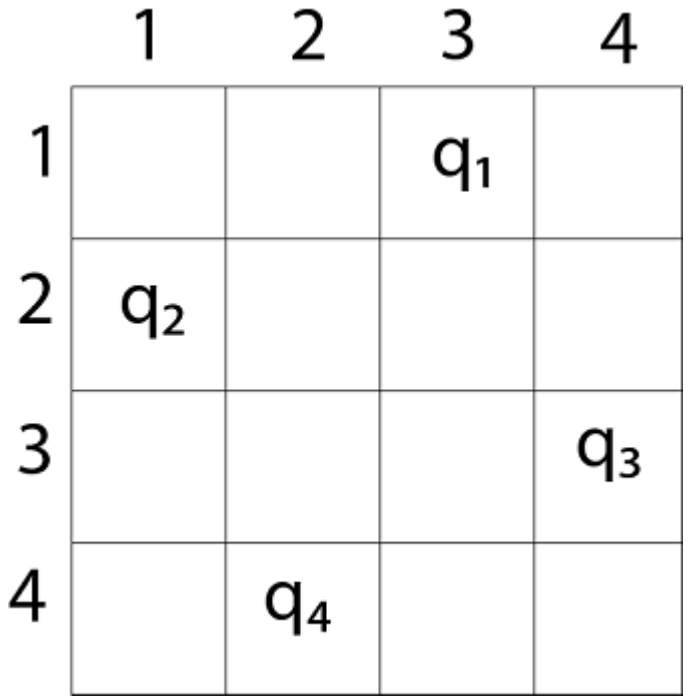
# 4*4 Queen Problem Solution

Solution



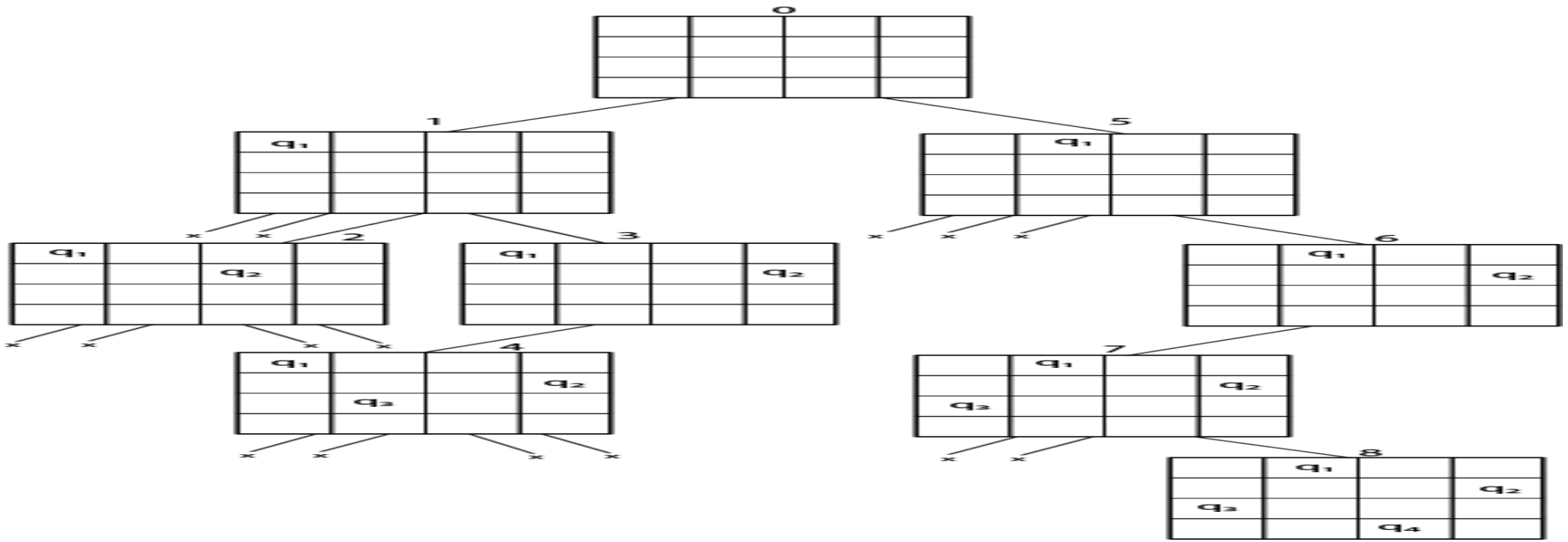
# 4*4 Queen Problem Solution

- This is the mirror image solution of 4*4 queen problem



# 4*4 Queen Problem Solution

The implicit tree for 4 - queen problem for a solution (2, 4, 1, 3) is as follows:



# Branch and Bound

- Where backtracking uses a depth-first search with pruning, the branch and bound algorithm uses a breadth-first search with pruning
- Branch and bound uses a queue as an auxiliary data structure
- The Branch and Bound Algorithm:- Starting by considering the root node and applying a lower-bounding and upper- bounding procedure to it • If the bounds match, then an optimal solution has been found and the algorithm is finished If they do not match, then algorithm runs on the child nodes
- Efficiency of Branch and Bound:- In many types of problems, branch and bound is faster than branching, due to the use of a breadth-first search instead of a depth-first search • The worst case scenario is the same, as it will still visit every node in the tree

# Thank You