UNIT V – BRANCH AND BOUND AND BACKTRACKING

Backtracking: N-Queens problem - Hamiltonian cycles – Graph coloring – Sum of subset. Branch and bound: The method – FIFO branch and bound- LC branch and bound – 0/1 Knapsack problem - Traveling salesman problem.

BACKTRACKING

Backtracking algorithms are like problem-solving strategies that help explore different options to find the best solution. They work by trying out different paths and if one doesn't work, they backtrack and try another until they find the right one. It's like solving a puzzle by testing different pieces until they fit together perfectly.

How Does a Backtracking Algorithm Work?

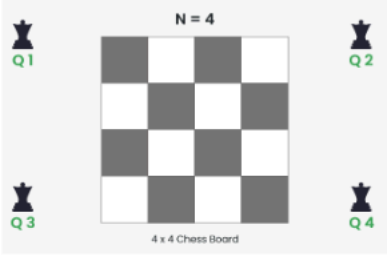
A backtracking algorithm works by recursively exploring all possible solutions to a problem. It starts by choosing an initial solution, and then it explores all possible extensions of that solution. If an extension leads to a solution, the algorithm returns that solution. If an extension does not lead to a solution, the algorithm backtracks to the previous solution and tries a different extension.

The following is a general outline of how a backtracking algorithm works:

1. Choose an initial solution.
2. Explore all possible extensions of the current solution.
3. If an extension leads to a solution, return that solution.
4. If an extension does not lead to a solution, backtrack to the previous solution and try a different extension.
5. Repeat steps 2-4 until all possible solutions have been explored.

N-QUEENS PROBLEM

The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other.



For example, the following is a solution for the 4 Queen problem.

Implementation

Follow the steps mentioned below to implement the idea:

- Start in the leftmost column
- If all queens are placed return true
- Try all rows in the current column. Do the following for every row.
    - If the queen can be placed safely in this row
        - Then mark this [row, column] as part of the solution and recursively check if placing queen here leads to a solution.
        - If placing the queen in [row, column] leads to a solution then

return true.
        - If placing queen doesn't lead to a solution then unmark this [row, column] then backtrack and try other rows.
    - If all rows have been tried and valid solution is not found return false to trigger backtracking.

Example: [Refer class notes]
Time Complexity: O(N!)
Auxiliary Space: O(N)

HAMILTONIAN CYCLES

Hamiltonian Cycle or Circuit in a graph G is a cycle that visits every vertex of G exactly once and returns to the starting vertex. If a graph contains a Hamiltonian cycle, it is called a Hamiltonian graph, otherwise it is non-Hamiltonian.

Finding a Hamiltonian Cycle in a graph is a well-known NP-complete problem, which means that there's no known efficient algorithm to solve it for all types of graphs. However, it can be solved for small or specific types of graphs.

The Hamiltonian Cycle problem has practical applications in various fields, such as logistics, network design, and computer science.