TAGORE ENGINEERING COLLEGE

Approved by AICTE, New Delhi | Affiliated to Anna University, Chennai
Rathinamangalam, Vandalur – Kelambakkam Road, Chennai, Tamil Nadu- 600 127, India

Department of Information Technology

ACTIVITY: CASE STUDY

COURSE NAME: CD 3291 Data Structures and Algorithms **DATE:20.10.2022**

YEAR/SEMESTER: II/III

Faculty: Ms.G.Bhuvaneshwari

Topic: Backtracking algorithm

Case studies revolve around real-life problems, serving as a platform to highlight crucial points and address concerns. This approach enhances student's innovative learning and idea generation, motivating them to apply the skills they have acquired. Unlike traditional lectures, case studies involve complete student participation across a broader range of skills.

Backtracking is a powerful algorithmic technique used for solving problems incrementally by trying partial solutions and then abandoning them if they are not valid. It is often used for combinatorial problems, puzzles, and constraint satisfaction problems, such as:

- **N-Queens problem**
- **Sudoku**
- **Hamiltonian path/cycle**
- **Subsets and combinations generation**

Backtracking Approach

This approach rejects all further moves if the solution is declined at any step, goes back to the previous step and explores other options.

Algorithm

Let's go through the steps below to understand how this algorithm of solving the 8 queens problem using backtracking works:

- **Step 1:** Traverse all the rows in one column at a time and try to place the queen in that position.
- **Step 2:** After coming to a new square in the left column, traverse to its left horizontal direction to see if any queen is already placed in that row or not. If a queen is found, then move to other rows to search for a possible position for the queen.
- **Step 3:** Like step 2, check the upper and lower left diagonals. We do not check the right side because it's impossible to find a queen on that side of the board yet.
- **Step 4:** If the process succeeds, i.e. a queen is not found, mark the position as '1' and move ahead.

- **Step 5:** Recursively use the above-listed steps to reach the last column. Print the solution matrix if a queen is successfully placed in the last column.
- **Step 6:** Backtrack to find other solutions after printing one possible solution.

Tagore Engineering College
Department of Information Technology

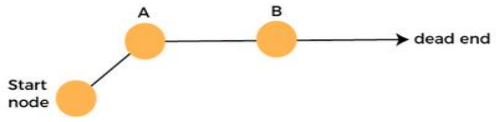
Name of the Student: K V V Mathimithra
Year/Semester: Second/III
Date:20.10.2022
Subject Code: CS3291-Data Structures and Algorithms
Topic: Case Study on Backtracking Algorithms

Backtracking is one of the techniques that can be used to solve the problem. We can write the algorithm using this strategy. It uses the Brute force search to solve the problem, and the brute force search says that for the given problem, we try to make all the possible solutions and pick out the best solution from all the desired solutions. This rule is also followed in dynamic programming, but dynamic programming is used for solving optimization problems. In contrast, backtracking is not used in solving optimization problems. Backtracking is used when we have multiple solutions, and we require all of those solutions.

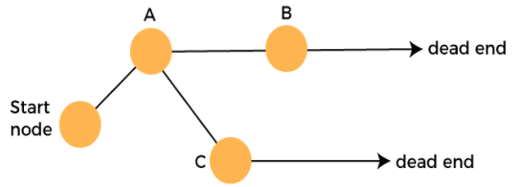
The backtracking name suggests that we are going back and coming forward; if it satisfies the condition, then return success, or we go back again. It is used to solve a problem in which a sequence of objects is chosen from a specified set so that the sequence satisfies some criteria.

Backtracking is a systematic method of trying out various sequences of decisions until you find out that works. Let's understand through an example.

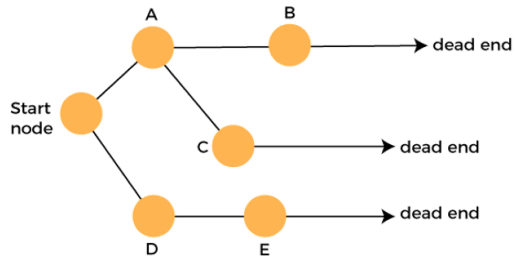
We start with a start node. First, we move to node A. Since it is not a feasible solution so we move to the next node, i.e., B. B is also not a feasible solution, and it is a dead-end so we backtrack from node B to node A.



Suppose another path exists from node A to node C. So, we move from node A to node C. It is also a dead-end, so again backtrack from node C to node A. We move from node A to the starting node.



Now we will check any other path exists from the starting node. So, we move from start node to the node D. Since it is not a feasible solution so we move from node D to node E. The node E is also not a feasible solution. It is a dead end so we backtrack from node E to node D.



8-Queens Problem

Problem Statement

Given an 8x8 chess board, you must place 8 queens on the board so that no two queens attack each other. Print all possible matrices satisfying the conditions with positions with queens marked with '1' and empty spaces with '0'. You must solve the 8 queens problem using backtracking.

Note 1: A queen can move vertically, horizontally and diagonally in any number of steps.

Note 2: You can also go through the N-Queen Problem for the general approach to solving this problem.

Sample Example

Example: One possible solution to the 8 queens problem using backtracking is shown below. In the first row, the queen is at E8 square, so we have to make sure no queue

n is in column E and row 8 and also along its diagonals. Similarly, for the second row, the queen is on the B7 square, thus, we have to secure its horizontal, vertical, and diagonal squares. The same pattern is followed for the rest of the queens.

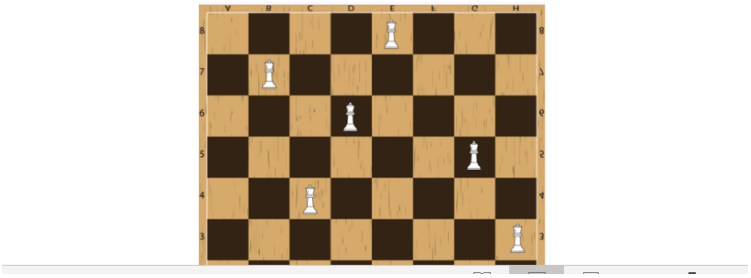


Figure 2 Case study on Backtracking algorithms