Data Structures and Algorithms

Lecture 12

"Human mind is subject to the law of cause and effect. IF not, THEN you have no idea about IF-THEN algorithm."

— Toba Beta, Master of Stupidity

Backtracking - introduction

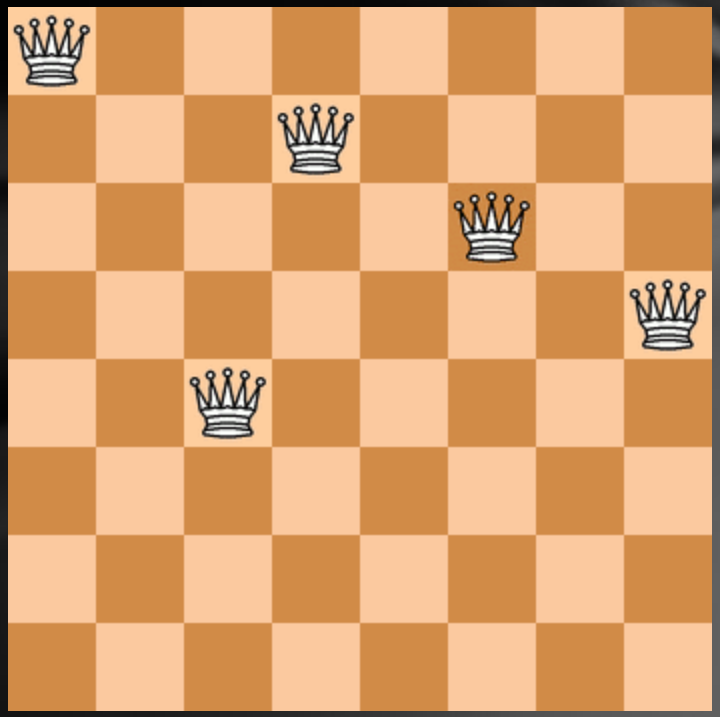
- Backtracking is an algorithm that is used to find all solutions to a problem.
- Partial solutions are discarded early if they can not possibly be completed to a valid solution.
- Backtracking is widely used, even though it has poor performance - O(n!) - since there are problems that can not be solved in any other way
- Examples:
  1. Given a chess board of N x N squares, find all possible solutions for placing N queens on the board without any two queens attacking each other.
  2. Find all permutations of n numbers.
  3. Given a connected, undirected graph, find the shortest route that visits all edges (Chinese postman problem).
  4. Given a connected, undirected graph, find the shortest cycle that visits all vertices exactly once (traveling salesman problem).

Backtracking - algorithm

- Given a set of values S = {s_0, s_1, s_2, … s_n}, a solution to the problem is a valid sequence of values from S.
- Backtracking uses a stack
- Algorithm:
  - Initialize stack with the element before s_0
  - While the stack isn't empty, repeat:
    - While the element on the top of the stack has a successor and it is NOT valid, find the next successor
    - If the successor is valid and the stack is a valid solution, then use the result; continue
    - If the successor is valid and the stack is not yet a valid solution, push the element before s_0 onto the stack, increasing it by one; continue.
    - If there is no successor for the current stack head, pop the element from the stack; continue

Backtracking – Queen problem

- The stack contains the column index for the queen on the row given by the position in the stack (E.g.: if there is 3 on the first position in the stack – 0 – it means that a queen is places on column 3 and row 0.
- A new stack position is initialized with -1
- A position has a successor if the value on that position is less than 7 (or generally N-1, for an N x N board)
- A successor is valid if by placing a queen at that position, it doesn't attack any queens already on the stack
- A stack is a valid solution if there are N queens placed on the stack



Backtracking – Queen problem

Exercise: Implement a backtracking algorithm that solves the queen problem for a generic N x N board.

Backtracking – Permutation problem

- Problem: write a program that will print all possible permutations of numbers from 1 to N.

- The stack contains the permuted numbers
- A new stack position is initialized with 0
- A position has a successor if the value on that position is less than N
- A successor is valid if the value is not yet present on the stack
- A stack is a valid solution if there are N numbers on the stack

- Exercise: Write a backtracking algorithm to solve this problem
- Exercise: Write a backtracking algorithm to print all sorted combinations of N by K numbers.
- Exercise: Rewrite the permutation algorithm using recurrence.

Thank you!