Backtracking

- Another strategy for recursion
- When algorithm needs to choose between multiple alternative steps, recursively evaluate *each* one and then choose the best.

N Queens

Place n queens on $n \times n$ chessboard, so that no two queens are attacking each other (occupying same row, column, or diagonal).

Recursive solution:

1. start with first row, proceed left to right.
2. To place the rth queen, try all squares in row r from left to right
3. If a particular square can be attacked by already placed queens, pass
4. else, place queen on that square and recurse.

Nice property of this solution: can generate *all* possible solutions.

How to efficiently represent board state?

Consider an array $Q[1..n]$, where $Q[i]$ stores the column index of the queen in row i.

(Show algorithm)

For this algorithm, we can build a nice recursion tree:

- Each node corresponds to a subproblem (partial solution)
- Root is the empty board
- Leaves correspond to solutions that cannot be extended, either because there is a queen on every row (solved), or because there are no valid positions on the next row, in which case the recursion terminates and backtracks.

Does this remind us of any algorithms? (DFS)

(Show complete recursion tree for n=4)

Game trees

- Generalized representation for game states
- Game theory!
- Intro to sugar-packet game

State of a game:

- locations of all pieces
- identity of current player
- States are connected in a game tree

- Game tree has an edge from state x to state y if and only if the current player in state x can legally move to state y.
- Root is initial state of game
- Every path from root to leaf is a complete game.
- Game state is *good* if either current player has already won, or if current player can move to a bad state for opposing player.
- Game state is *bad* if either the current player has already lost, or if every available move leads to a good state for the opposing player.
- Generalization: node in the tree is good if it has at least one bad child, and a node is bad if all its children are good.

(Show algorithm)

Text Segmentation

- "BOTHEARTHANDSATURNSPIN" -> "Both earth and saturn spin" or "bot heart hands at urn spin"
- Given a string of characters, can it be segmented into english words?
- Given subroutine `IsWord(w)`
- Strategy: consume sequence of characters one at a time, decide where to place separation
- Again, use recursive brute force
- Note on representations and notation for algorithms acting on arrays
- Show both algos

Analysis

Running time of splittable:

$$T(n) \leq \sum_{i=0}^{n-1} T(i) + O(n)$$

To solve:

- Replace O(n) with αn
- Assume we make every recursive call (conservative)
- Form T(n) and T(n-1), subtract, get $T(n) = 2T(n-1) + \alpha$