Backtracking and Memoization

Lecture 12
Tuesday, October 6, 2020
12.1
On different techniques for recursive algorithms
Recursion

Reduction:
Reduce one problem to another

Recursion
A special case of reduction
1. reduce problem to a smaller instance of itself
2. self-reduction

- Problem instance of size $n$ is reduced to one or more instances of size $n - 1$ or less.
- For termination, problem instances of small size are solved by some other method as base cases.
Recursion in Algorithm Design

1. **Tail Recursion**: problem reduced to a single recursive call after some work. Easy to convert algorithm into iterative or greedy algorithms. Examples: Interval scheduling, MST algorithms, etc.

2. **Divide and Conquer**: Problem reduced to multiple independent sub-problems that are solved separately. Conquer step puts together solution for bigger problem. Examples: Closest pair, deterministic median selection, quick sort.

3. **Backtracking**: Refinement of brute force search. Build solution incrementally by invoking recursion to try all possibilities for the decision in each step.

4. **Dynamic Programming**: problem reduced to multiple (typically) dependent or overlapping sub-problems. Use memoization to avoid recomputation of common solutions leading to iterative bottom-up algorithm.
THE END

... (for now)