Algorithm paradigms

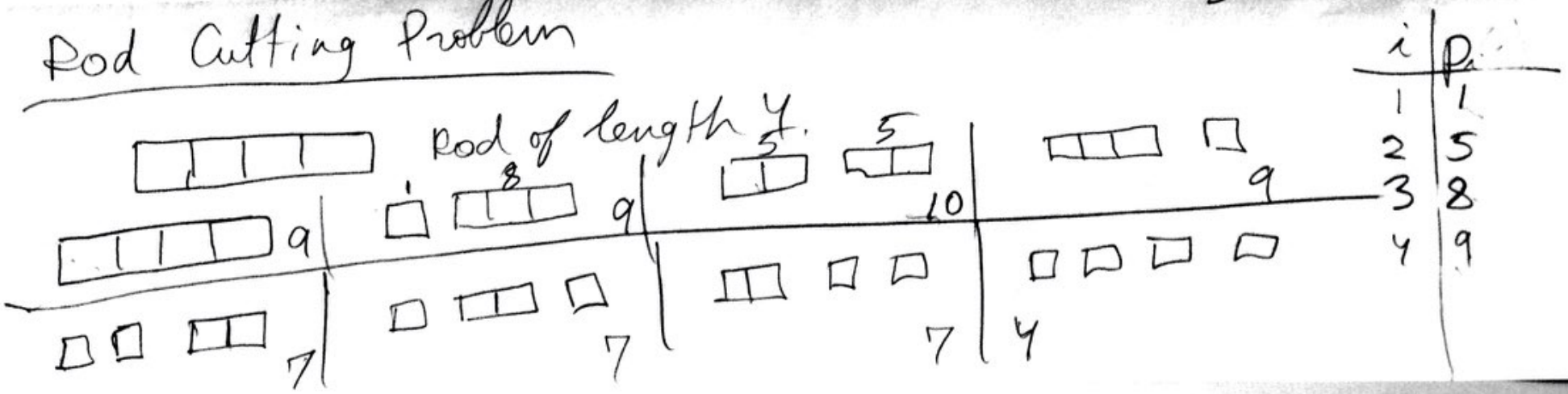
1. Incremental algorithms. (e.g. Ins. Sort).
2. Divide and Conquer (e.g. BinarySearch, MergeSort, Max-subarray).
3. Dynamic Programming.
   misnomer

Optimization Problem.

1. Feasible solution
2. Cost of a solution.

We want to find a feasible with the lowest cost
minimization or maximizations.

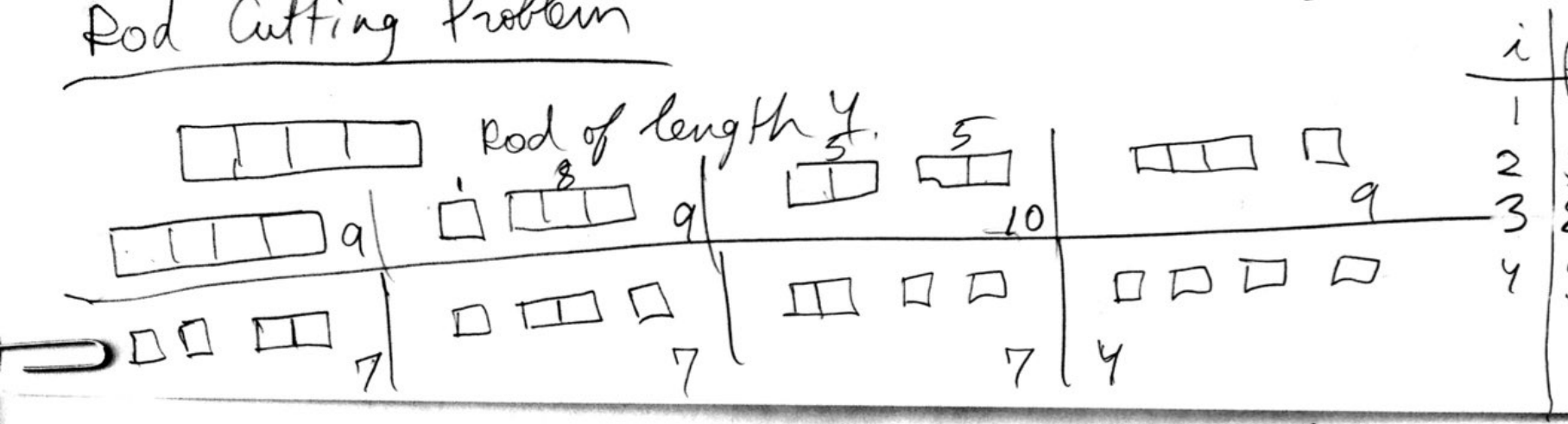
Rod Cutting Problem



i	p_i
1	1
2	5
3	8
4	9

Rod Cutting Problem

maximizations.



i	p_i
1	1
2	5
3	8
4	9

- 2^{n-1} ways to cut the rod: 2^{n-1} feasible solutions.
- The Input:
  - a length n and a table of prices $p_1, p_2, \ldots, p_n$.
    (p_i is the price for a rod of length i).
- The output: The max revenue obtainable for rods whose lengths sum to n.
- Brute Force alg: Run time: $\Theta(2^n \cdot n)$ exponential time.
- Denote r_i to be the max revenue for a rod of length i

Dynamic Programming:

- Overlapping subproblems.
- Idea: solve each subproblem just once, and store the result in a table. Next time, just look up the solution in the table.
- two approaches: memoization, bottom-up.

Memoization

```
Mem-Cut-Rod(p,n)
    Let r[0..n] be a new array initialized to -∞
    return Mem-Cut-Rod-Aux(p,n,r)
```