6.00²

Lecture 2: Optimization Problems.

Pros of greedy algorithm and cons

- Easy to implement
- Computationally efficient.

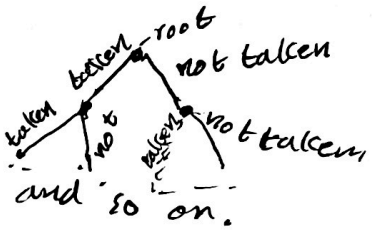
- not ~~always~~ always gives the optimum solution.
- we don't know how close it's to optimum.

Brute force & search tree

implement 0/1 knapsack

- using tree is buit top down starting with root.
- then build left branch of consequesce of taking item
- right branch of not taking item
- keep this process while knapsack is not full
- Lastly take the branch with most value.

example



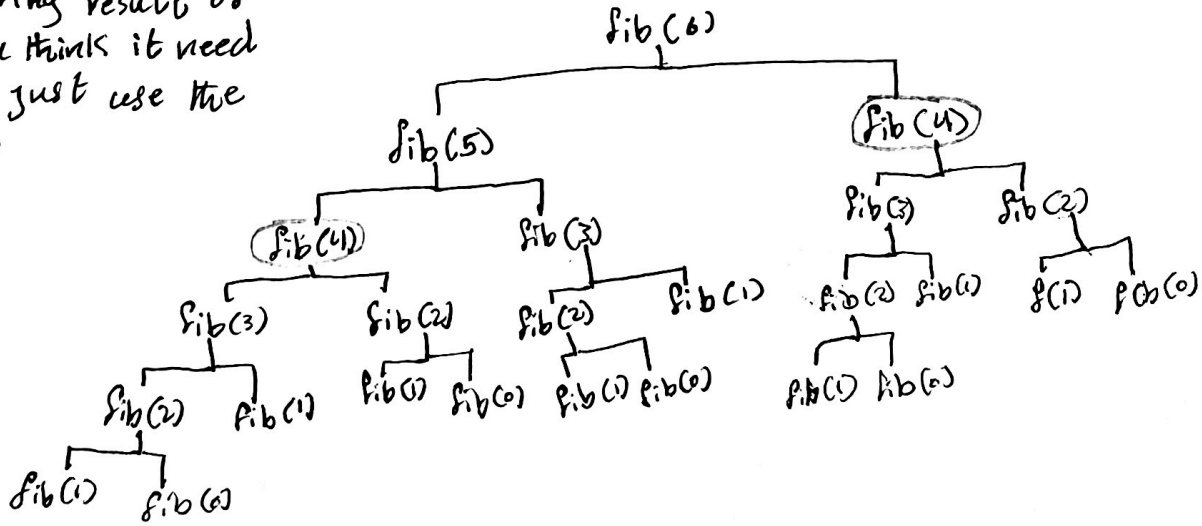
Complexity

- number of nodes at depth i is 2^i
- number of total nodes $= \sum_{i=0}^{n} 2^i = O(2^{n+1})$

Dynamic Programming

memoization is storing result of something that you thinks it need to compute but just use the stored result.

Call tree of fib



most of the subtree is replica.

memoization work only when problem have

1: Optimal substructure = "a globally optimal solution can be found by combining optimal solutions to local subproblems"

2: overlapping subproblems. "finding optimal solution involves solving the same problem multiple times"

Dynamic programming can be used to solve 0/1 knapsack.