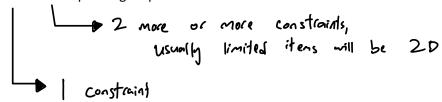
Dynamic Programming

- Solving problems efficiently by reusing solutions that has already been computed.

Solving DP:

- 1. Solve the naïve way
 - without time/space complexity first • Brute force it without thinking of time complexity -- permutations
- 2. Optimal Substructure & Overlapping Subproblems
 - How to construct best solution given a iteration. Optimal Substructures Overlapping Subproblems
- repeated solution 3. Top-Down vs Bottom Up approach
 - · Not really important
- 4. Constraints
 - What limitations are imposed when constructing the optimal substructure Eg. min no. of coins, max. value of items, limited no. of items etc
- 5. Base Case
- 6. Memoization Array
 - 1D or 2D depending on problem statement constraints



Rod Cutting Problem

Given a rod of length n inches and an array of prices that includes prices of all pieces of size smaller than n. Determine the maximum value obtainable by cutting up the rod and selling the pieces.

Length	1	2	3	4	5	6	7	8
Price	1	5	8	9	10	17	17	20

Value = 22, 2 + 6

Length	1	2	3	4	5	6	7	8
Price	3	5	8	9	10	17	17	20

options pleas plianel, meno Li options Total	
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	of ption

memo		1	2	3	4	5	16	7	2
\$\$	0	1	5	8	(0)	13	17	18	22

option	preu options!	total
141	C37: \$8	\$9
2 \$5	C2] = \$5	\$ 10 A Max
318	[(]=\$)	\$ 9
F)\$9	Co7 = \$0	\$ 9

Partition Problem

Given a set of non-negative integers, find if we can partition it into two subsets such that the sum of elements in both the subsets are equal.

[5,8,3,1,4,7] 28//2=[4]
Each cell has ① exclude, [i-1][j] lst = [1, 2, 3, 4]TRUE (1, 4) & {2, 3} lst = [1, 1, 3, 4, 7]TRUE {1,3,4} & {1,7} lst = [2, 3, 4, 6]**FALSE** # each cell represent a 2 include, [i-1] [i-option]] possible combination 14 13 12 10