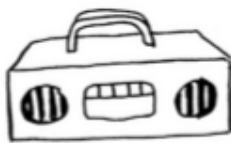


DYNAMIC PROGRAMMING

- TECHNIQUE
- SOLVE PROBLEM => SOLVE SUBPROBLEM
- OPTIMIZATION PROBLEM
Time, space, benefits, etc, ...

KNAPSACK PROBLEM



STEREO
\$3000
4 lbs



LAPTOP
\$2000
3 lbs



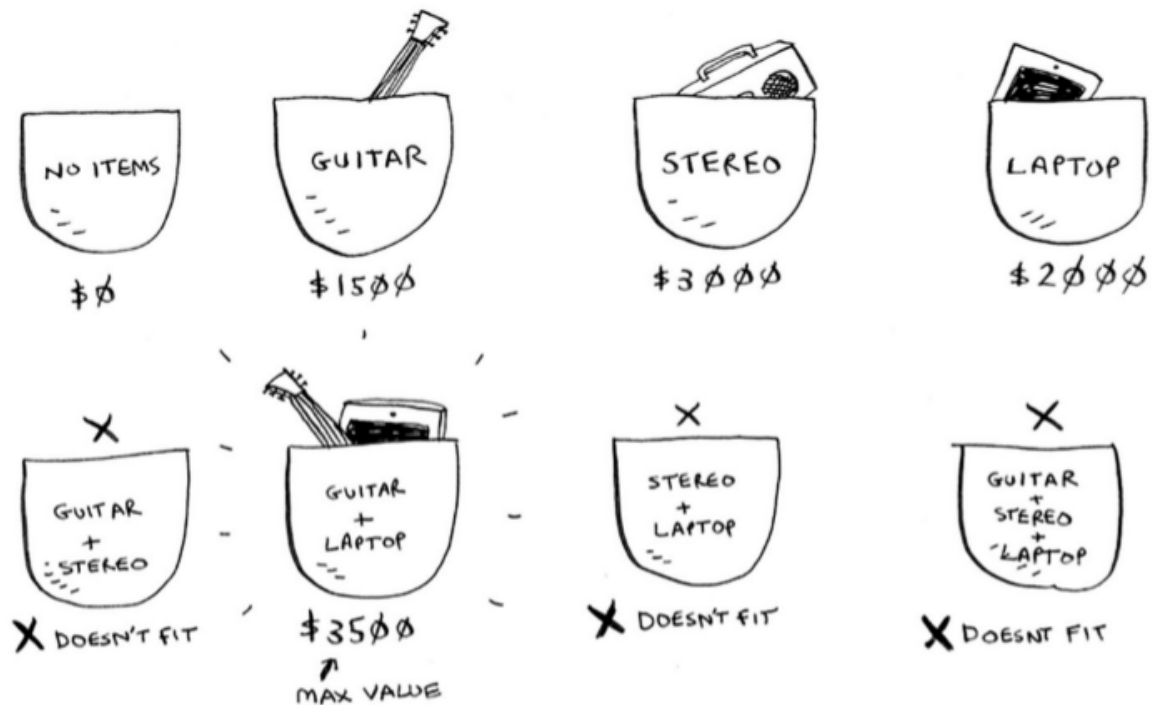
GUITAR
\$1500
1 lbs

MAX CAPACITY = 4 lb

MAX BENEFIT = ?????

SIMPLE SOLUTION

try every possible set of goods



TAKE TIME !!! $\Rightarrow O(2^n)$

32 items = \sim 4 BILLION
POSSIBLE SETS

BEST SOLUTION

Dynamic Programming

(bottom up solution)

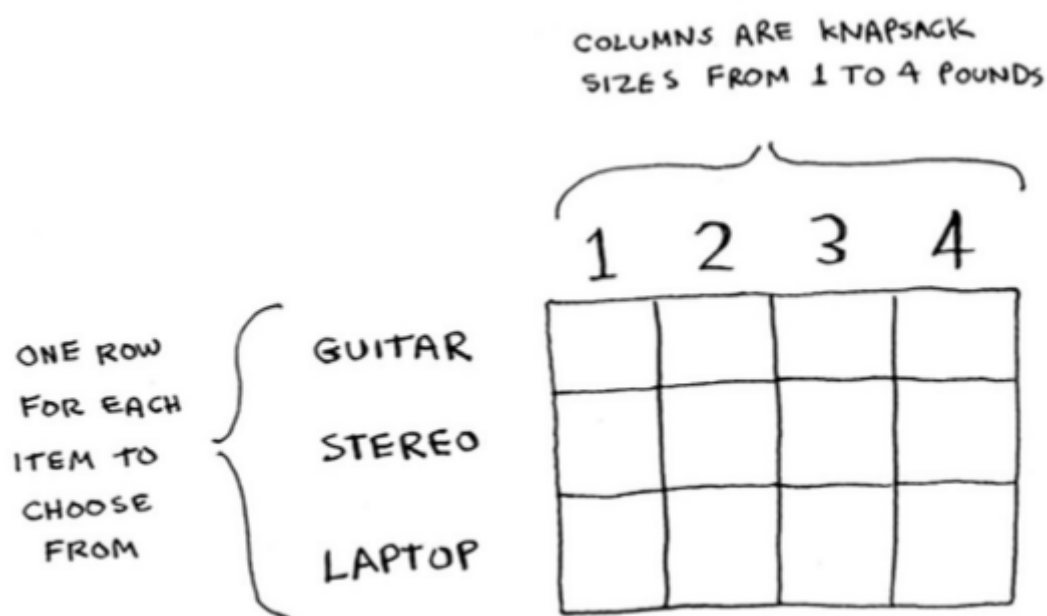
Solving :

problem => **subproblem**

then

knapsack => **sub-knapsack**

Grid :



$$\begin{array}{c} \text{ROW} \quad \text{COLUMN} \\ \downarrow \quad \downarrow \\ \text{CELL}[i][j] \end{array} = \max \text{ of } \left\{ \begin{array}{l} 1. \text{ THE PREVIOUS MAX (VALUE AT CELL } [i-1][j]) \\ \text{VS} \\ 2. \text{ VALUE OF CURRENT ITEM + VALUE OF THE REMAINING SPACE} \\ \quad \quad \quad \uparrow \\ \quad \quad \quad \text{CELL}[i-1][j - \text{ITEM'S WEIGHT}] \end{array} \right.$$

	1	2	3	4
GUITAR	\$1500 ↓ G	\$1500 ↓ G	\$1500 ↓ G	\$1500 G
STEREO	\$1500 ↓ G	\$1500 ↓ G	\$1500 G	\$3000 S
LAPTOP	\$1500 G	\$1500 G	\$2000 L	\$3500 L G

↑
THE ANSWER!

$O(n^2)$

What happen if we add item ?

	1	2	3	4
GUITAR	\$1500 G	\$1500 G	\$1500 G	\$1500 G
STEREO	\$1500 G	\$1500 G	\$1500 G	\$3000 S
LAPTOP	\$1500 G	\$1500 G	\$2000 L	\$3500 LG
<u>IPHONE</u>				

↑
NEW ANSWER

\$1500 G	\$1500 G	\$1500 G	\$1500 G
\$1500 G	\$1500 G	\$1500 G	\$3000 S
\$1500 G	\$1500 G	\$2000 L	\$3500 LG
\$3500 I	\$3500 IG	\$3500 IG	\$4000 IL

↑
NEW
ANSWER

RECAP

- . D.P is usefull to **OPTIMIZE** given a **CONSTRAINT**
- . Use D.P if **PROBLEM** can be broken into **SUBPROBLEMS**
- . D.P solution always involve **GRID**
- **NO SINGLE FORMULA !**