0-1 Knapsack

① 0-1 Knapsack
② 2nd Change Prob

Given: list of items, each has weight, value

Item	Weight	Value
1	4	10
2	2	3
3	5	11
4	7	13

4/5/18 ①

Knapsack Weight Limit: 9 units

Problem what's the max value of any subset of items that weighs less than or equal to the weight limit?

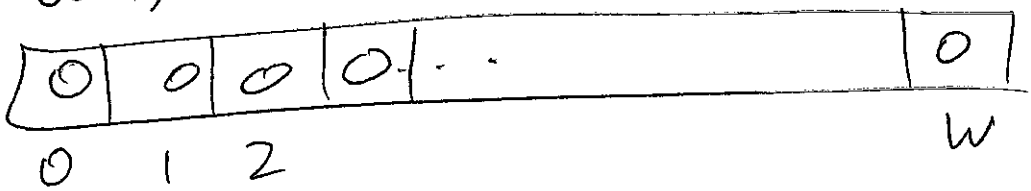
Alg #1

Divide + Conquer → try all subsets w/ Item 1
→ try all subsets w/o Item 1

runtime $O(2^n)$, n = # items

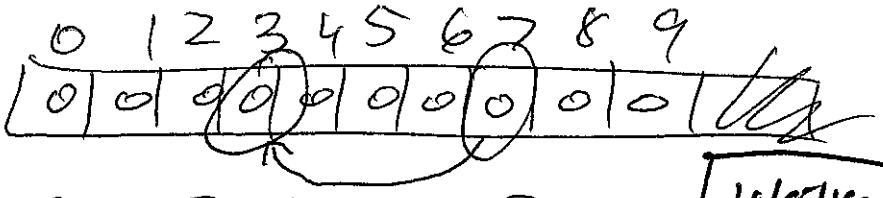
Alg #2 – Dynamic Programming Alg

max Weight = W



dp[i] = max value of any subset of items that weighs i of fewer units.

	W	V
I_1 →	4,	10
I_2	2,	3
I_3	5,	11
I_4	7,	13



4/5/18 ②

Can I_1 help us?
Currently I can hold 7 lbs.
and my best value is 0.
↳ If I took I_1, I only 3 lbs left
↳ new value $dp[3] + 10 = 10$

j = item #

$$dp[i] = \max(dp[i], dp[i - weights[j]] + values[j])$$

Don't take item j.

Do take item j.

We run inner loop backwards to prevent stealing > 1 copy of an item!

```
for (j = 0; j < n; j++) {
    for (i = max; i >= weights[j]; i--)
        dp[i] = Math.max(dp[i], dp[i-weights[j]] + values[j];
}
```

If forwards, allow > 1 copy

	W	V
I_1	4,	10
I_2	2,	3
I_3	5,	11
I_4	7,	13

	0	1	2	3	4	5	6	7	8	9
	0	0	0	0	0	0	0	0	0	0
I_1	0	0	0	0	10	10	10	10	10	10
I_2	0	0	3	3	10	10	13	13	13	13
I_3	0	0	3	3	10	11	13	14	14	21/I_3
I_4	0	0	3	3	10	11	13	14	14	21

Items	W	Value
CC granola	4	6
Mandarin	2	4
Cheese Stick	3	5
Chocolate	1	3
Bananas	6	9
CCPB granola	4	7

$W = 10$

	1	2	3	4	5	6	7	8	9	10
	0	0	0	0	0	0	0	0	0	0
CCg	0	0	0	6	6	6	6	6	6	6
mand	0	4	4	6	6	10	10	10	10	10
cheese	0	4	5	6	9	10	11	11	15	15
choc	3	4	7	8	9	12	13	14	15	18
Bananas	3	4	7	8	9	12	13	14	16	18
CCPB granola	3	4	7	8	10	12	14	15	16	19

Change Problem

Given: Denominations

1, 2, 4, 7, 12

What is the fewest # of coins necessary to make change for n cents?

dp[i] = fewest # of coins to make change for i cents

init dp[i] = i (use pennies)

rest of the coins

dp[i] = min(dp[i], dp[i - denom[j]] + 1)

dp[74] = min(dp[74], dp[74-25] + 1)

dp[i - denom[j]] + 1: fewest coins to make change for i - denom[j] cents. + 1 new coin j