0-1 Knapsack

Inner tube: 106g	26 troy dr $8
trading cards: 38g	10 dr $25
drink coupons: 2g	1 dr $1
book: 246g	50dr $5
tp: 164g	41dr $50

Jordan and Jamie
carbon fiber reinforced titanium magnesium rods
Avery and Ashley

Problem: Given 1) n items with weights $w_1, \ldots, w_n$ and values $v_1, \ldots, v_n$ (positive integers)

2) weight bound W

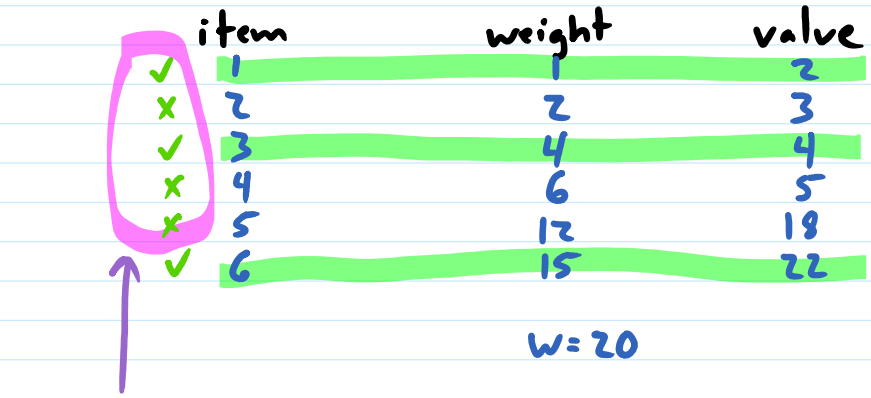
Find subset of n items of highest total value and total weight $\leq W$

Brute force $O(n2^n)$:(

		weight (troy dr)	value ($)
1)	inner tube	26	8
2)	trading cards	10	25
3)	book	50	5
4)	drink coupons	1	1
5)	toilet paper	41	50

$W = 50$

0-1 Knapsack Optimal Substructure



total = 28

the highest valued subset of 1st 5 items with total weight ≤ 5 → Suppose that {1,3,6} is optimal for n = 6, W = 20 but {1,3} is not optimal for n = 5, W = 5

Then we could do better than 6 for n = 5, W = 5 and better than 6+22 for n = 6, W = 20

So {1,3,6} would not be optimal

	item	weight	value
✓	1	1	2
x	2	2	3
✓	3	4	4
x	4	6	5
x	5	12	18
✓	6	15	22

total = 28

W = 20

the highest valued subset of 1st 5 items with total weight ≤ 5

$OPT[6,20] = v_6 + OPT[5, 20 - w_6]$

$OPT[i,j]$ = max possible value of subset of 1st i items with total weight $\leq j$ =

$$\begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ OPT[i-1,j] & \text{if } w_i > j \\ \max\left(v_i + OPT[i-1, j-w_i],\ OPT[i-1,j]\right) & \end{cases}$$

include i / don't include i

```
for j = 0 to W
    OPT[0,j] = 0
for i = 0 to n
    OPT[i,0] = 0
for i = 1 to n
    for j = 1 to W
        OPT[i,j] =
```

O(nW) entries
O(1) each

O(nW)

OPT

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	0	2	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	0	2	3	5	5	6	7	9	9	9	9	9	9	9	9	9	9	9	9	9	9
4	0	2	3	5	5	6	7	9	9	10	10	11	12	14	14	14	14	14	14	14	14
5	0	2	3	5	5	6	7	9	9	10	10	11	18	20	21	23	23	24	25	27	27
6	0	2	3	5	5	6	7	9	9	10	10	11	18	20	21	23	24	25	27	27	28

pseudopolynomial : $O(n \cdot 2^{\#\text{bits in } W})$

1 3 6

CHOOSE

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
2	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
3	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
4	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
5	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
6	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y

```
for i = 1 to n
  for j = 1 to W
    if OPT[i,j] = OPT[i-1,j]
      CHOOSE[i,j] = N
    else
      CHOOSE[i,j] = Y
```