

CS 252

F, 10 May 2024

$$T(n) \leq T(n-1) + n; \quad T(1) = 1$$

Guess

$$T(2) = T(1) + 2 = 3 \quad 1+2$$

$$T(3) = T(2) + 3 = 6 \quad 1+2+3$$

$$T(4) = T(3) + 4 = 10 \quad 1+2+3+4$$

⋮

$$T(n) \leq \underbrace{n(n+1)}_{2}$$

Prove by induction $T(n) \leq n(n+1)$

Base case : $T(1) = 1 \leq \frac{1(1+1)}{2} = 1$ ✓ P_1 is true

Engine

Proposition
parameterized by k

$$P_k : T(k) \leq \frac{k(k+1)}{2}$$

If P_k is true, then P_{k+1} must be true

If P_k then

$$T(k) \wedge T(k+1)$$

so

$$\begin{aligned} T(k+1) &\wedge T(k) + k+1 \\ &\wedge \frac{T(k+1)}{2} + k+1 \\ &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

i.e.
 P_{k+1} is
true

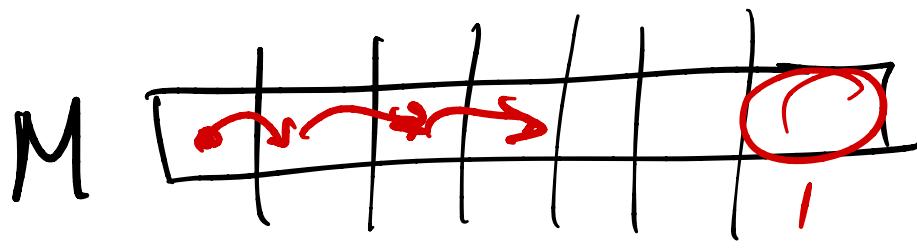
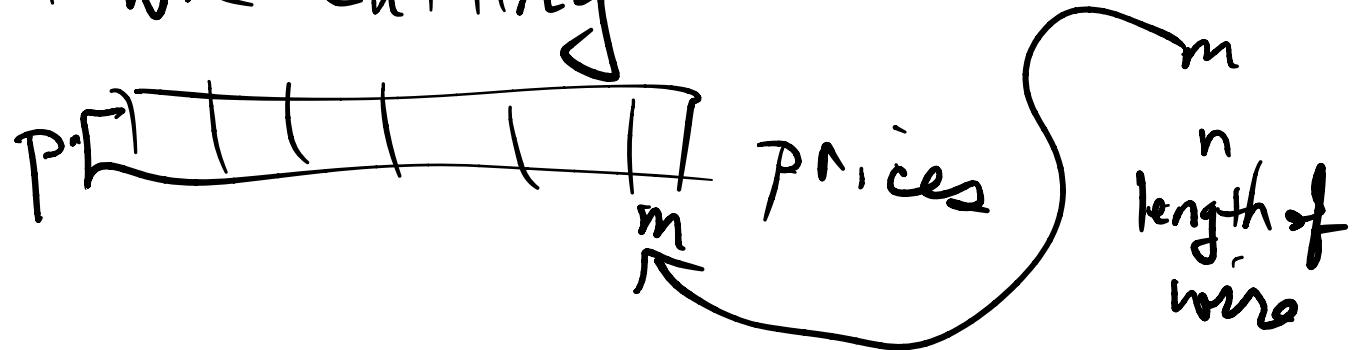
I've shown

$\neg A$ is true

$P_k \Rightarrow P_{k+1} \neq k$

done.

Recall wire-cutting



$M[n]$ was
the answer

Knapsack Problem

sizes	0	1	2	3	4	5	6	7
	2	5	1	1	3	1	8	6

values	7	6	1	2	4	9	10	6

capacity 12

Goal: maximize total value
keep total size ≤ 12

Take the diamonds,
not the piano

Robbing the house
Backpack can hold total size of 12.

What do you steal?

	0	1	2	3	4	5	6	7
sizes	2	5	1	1	3	1	8	6

values	7	6	1	2	4	9	10	6
--------	---	---	---	---	---	---	----	---

Capacity
12

Let $M(k, C) =$ the max value you can get
with objects $0 \dots k$ + capacity C

Goal $M(7, 12)$

	0	1	2	3	4	5	6	7
sizes	2	5	1	1	3	1	8	6
values	7	6	1	2	4	9	10	6
capacity	12							

$$\begin{aligned}
 & M(7, 12) \\
 &= \max \begin{cases} 6 + M(6, 6) \\ M(6, 12) \end{cases}
 \end{aligned}$$

$$M(k, C) = \max \begin{cases} \text{values}[k] + M(k-1, C - \text{size}[k]) \\ M(k-1, C) \end{cases}$$

Either we include item k in our haul, or we don't. Base case?

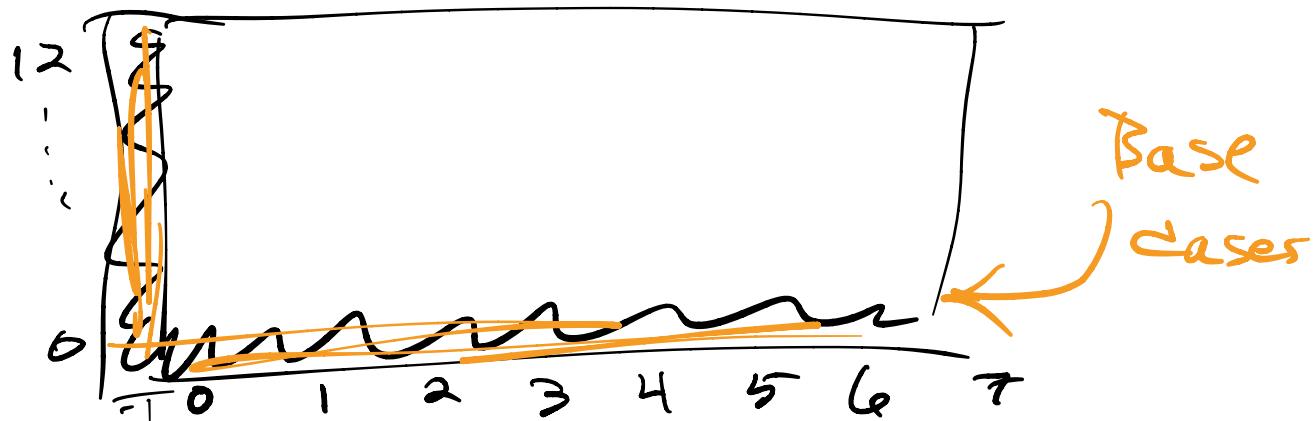
sizes	0	1	2	3	4	5	6	7
	2	5	1	1	3	1	8	6

values	7	6	1	2	4	9	10	6

Capacity 12

$$M(k, C) = \max \begin{cases} \text{values}[k] + M(k-1, C - \text{size}[k]) \\ M(k-1, C) \end{cases}$$

Base Case



	0	1	2	3	4	5	6	7
sizes	2	5	1	1	3	1	8	6

	7	6	1	2	4	9	10	6
values								

Capacity 12

$$M(k, C) =$$

$$\max \left\{ \begin{array}{l} \text{IF } C \geq \text{size}[k] \\ \text{value}[k] + M(k-1, C - \text{size}[k]) \\ M(k-1, C) \end{array} \right\}$$

$M(k, c)$ C  k

	0	1	2	3	4	5	6	7	8	9	10	11	12
-1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	7	7	7	7	7	7	7	7	7	7	7
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0

 $M(0, 1)$

$$= \max(M(-1, 1),$$

$\text{value}[0] + M(-1, 2)$

~~$7 + M(-1, 2)$~~

 ~~$M(-1, 2)$~~

$$M(k, c) = \max(M(k-1, c), \underbrace{\text{value}[k] + M(k-1, c - \text{size}[k])}_{\text{if } c \geq \text{size}[k]})$$

$M(k, c)$  k

		C												
		0	1	2	3	4	5	6	7	8	9	10	11	12
k	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	7	7	7	7	7	7	7	7	7	7	7
1	0	0	7	7	7	7	7	7	13	13	13	13	13	13
2	0	0	0	7	7	7	7	7	13	13	13	13	13	13
3	0	0	0	0	7	7	7	7	7	13	13	13	13	13
4	0	0	0	0	0	7	7	7	7	7	13	13	13	13
5	0	0	0	0	0	0	7	7	7	7	7	13	13	13
6	0	0	0	0	0	0	0	7	7	7	7	7	13	13
7	0	0	0	0	0	0	0	0	7	7	7	7	7	13

flow
long?

items

x

max capacity

$$M(k, c) = \max \left(M(k-1, c), \underbrace{\text{value}[k] + M(k-1, c - \text{size}[k])}_{\text{if } c \geq \text{size}[k]} \right)$$