CSE 102 2-13-20

---

continue ... coin changing

Initialization : $C[i, 0] = 0$

Ex $n = 4$, $d = (1, 3, 5, 6)$, $N = 8$

		j								
i	d_i	0	1	2	3	4	5	6	7	8
1	1	0	1	2	3	4	5	6	7	8
2	3	(0)	1	2	(1)	2	3	2	3	4
3	5	0	1	2	(1)	2	1	2	3	(2)
4	6	0	1	2	1	2	1	1	2	(2)

Back-track through table to solve part (2), optimal coin disbursement,

Ex. $n = 3$, $d = (2, 4, 5)$, $N = 8$

		j								
i	d_i	0	1	2	3	4	5	6	7	8
1	2	0	∞	1	∞	2	∞	3	∞	4
2	4	0	∞	1	∞	1	∞	2	∞	2
3	5	0	∞	1	∞	1	1	2	2	2

Recall:

$$C[i,j] = \min\left(C[i-1,j],\ 1 + C[i, j-d_i]\right)$$

Exercise (hard)

add another input: $l = (l_1, l_2, \ldots, l_n)$

where l_i is max # of coins of type d_i available.

Exercise

- Write a recursive algorithm that, given C (full, filled) table prints out an optimal coin disbursement.
- modify answer to print all optimal solutions

Problem Discrete Knapsack

A thief wants to steal n objects, labled $1, 2, \ldots, n$.

v_i = value of obj i

w_i = weight " " "

$(1 \le i \le n)$

Knapsack has capacity W

Goal: maximize value of goods stolen, without exceeding total weight W.

Define bitstring

$$x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$$

where

$$x_i = \begin{cases} 0 & \text{if obj } i \text{ not stolen} \\ 1 & \text{if } \ldots \ldots \text{ is } \ldots \end{cases}$$

Goal :

maximize : $\sum_{i=1}^{n} x_i \cdot v_i$ ← objective function

subject to : $\sum_{i=1}^{n} x_i \cdot w_i \leq W$

↑ constraint

Actually 2 Problems

(1) find max value that can be stolen.

(2) find which objects must be stolen to achieve value in (1).

Define

$V[i,j]$ = max value of objects $\{1,\dots,i\}$ whose total weight does not exceed j

have 2 alternatives for $V[i,j]$

- do not include object i:
  then $V[i,j] = V[i-1,j]$
- do include object i:
  then $V[i,j] = v_i + V[i-1, j-w_i]$

In general

$$V[i,j] = \max\left(V[i-1,j],\ v_i + V[i-1, j-w_i]\right)$$

including boundary & initialization

$$V[i,j] = \begin{cases} 0 & (i=0 \text{ \& } j \geq 0) \\ \max\left(V[i-1,j],\ v_i + V[i-1, j-w_i]\right) & (i>0 \text{ \& } j \geq 0) \\ -\infty & (j<0) \end{cases}$$

Range $\quad 1 \leq i \leq n$

$\quad\quad 0 \leq j \leq W$

Ex. $n = 5$,

$w = (1, 3, 5, 6, 7)$

$v = (1, 5, 12, 25, 30)$

$W = 10$

			j										
i	w_i	v_i	0	1	2	3	4	5	6	7	8	9	10
1	1	1	0	1	1	1	1	1	1	1	1	1	1
✓ 2	3	5	(0)	1	1	(5)	6	6	6	6	6	6	6
3	5	12	0	1	1	(5)	6	12	13	13	17	18	18
4	6	25	0	1	1	(5)	6	12	25	26	26	30	31
✓ 5	7	30	0	1	1	5	6	12	25	30	31	31	(35)

so $x = (0, 1, 0, 0, 1)$

Exercise: Write pseudo-code to fill this table.

Exercise

write recursive algorithm that backtracks through filled table prints which objects to steal.

Principle of Optimality
or Optimal Substructure

Any non-trivial instance of Problem is a combination of optimal solutions to some of its sub-instances

- coin change

$$c[i,j] = \min\left(c[i-1,j],\ 1 + c[i,j-d_i]\right)$$

- 0-1 Knapsack

$$V[i,j] = \max\left(V[i-1,j],\ v_i + V[i-1,j-w_i]\right)$$

Example

Shortest Paths in a Graph.

Defn Path, length of path

shortest u-v Path : $d(u,v)$

longest u-v Path : $\ell(u,v)$