CMPS 201   6-2-10

- hw 8 solutions Posted.
- hw 9 Review Problems.
  Do not turn in.
  Solutions included.
- 3 hour final on Mon June 7
- 6-8 Problems, Cumulative.
  emphasis on most recent
  material

Ex 0-1 Knapsack Problem:

A thief wishes to steal n objects, labeled $i = 1, \ldots, n$. Let

v_i = value of object i

w_i = weight . . . . . .

W = capacity (i.e. max weight) of Knapsack.

Goal: fill Knapsack so as to maximize total value while not exceeding W.

let

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

<u>Goal</u>:

maximize: $$\sum_{i=1}^{n} x_i v_i$$

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

where $v_i > 0$, $w_i > 0$, $W > 0$.

<u>Define</u>:

$$V[1 \ldots n;\ 0 \ldots W]$$

where

$V[i,j]$ = max value of objects in $\{1, \ldots, i\}$ whose total weight $\leq j$

$(1 \leq i \leq n,\ 0 \leq j \leq W)$

To Determine $V[i,j]$ we have 2 alternatives!

- Do not include obj i (even though you can.) In this case at most value $V[i-1,j]$ can be stolen.

- Include Obj i. This option increases value by v_i, and reduces capacity by w_i. So In this case the max value that can be stolen is

$$v_i + V[i-1, j-w_i]$$

So obviously

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

Including Boundry & out of bounds entries!

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

$\boxed{\text{Note: } V[i,0] = 0}$

Ex. $W = 10$, $n = 5$

i	w	v	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

Principle of Optimality:

- The optimal solution to any (non-trivial) instance is a combination of some of its sub-instance solutions.

- i.e. an optimal soln contains some optimal subproblem solns.

- In an optimal sequence of choices, each subsequence is also optimal.

We say: the problem exhibits optimal substructure.

Coin Change:

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

Knapsack:

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

hw 9: Review

- Canoe Rental Problem.
- Checkerboard Problem.

Ex Canoe Rental Problem

- have trading posts along a river: n posts as you travel down stream. Indexed $1, 2, \ldots, n$

we're given an array $R[i,j]$ defining cost of renting a canoe at post i, and returning it at post j, where $1 \leq i \leq j \leq n$. Assume $R[i,i] = 0$.

Assume $R[i,j] = \infty$ if $i > j$.

<u>Problem</u> Determine an optimal sequence of canoe rentals (i.e. lowest possible cost) starting at post 1 and ending at post n.

Really 2 problems

- determine cost of a cheapest sequence
- determine a cheapest seq.

Special case of S.P. in Graph