CSE 102 11-4-21

- Midterm 1 grading: expect done friday
- hw 4 solns posted
- hw 5 posted

Problem: Discrete Knapsack

A thief wants to steal n objects: $\{1, 2, \ldots, n\}$. For each $i = 1, \ldots, n$, let

v_i = value of obj. i

w_i = weight of obj. i

Thief has a knapsack with weight capacity $W \geq 0$.

Goal: steal max value of goods, subject to constraint (weight of goods) $\leq W$.

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

where

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

Goal: choose $x \in \{0,1\}^n$ so as to

- maximize: $\sum_{i=1}^{n} x_i v_i$ (objective function)

- subject to: $\sum_{i=1}^{n} x_i w_i \leq W$ (constraint)

Any bit string satisfying the constraint is called a feasible solution. an x satisfying both bullets is called an optimal solution.

As before we seek to

- find value of an opt. soln.
  (i.e. max value of objective fcn.)
- find opt. solution itself, i.e.
  a particular bit seq: $x = (x_1, \ldots, x_n)$

Create a table $V[1 \ldots n; 0 \ldots W]$

$V[i,j]$ = max value of a subset of $\{1, \ldots, i\}$ whose ~~total~~ weight is $\leq j$.

To fill cell (i,j) we 2 alternatives

(1) do not include obj. i (set $x_i = 0$)
at most $V[i-1, j]$ can be stolen

(2) include obj. i (set $x_i = 1$). This increases value by v_i and decreases capacity to $j - w_i$. Max value in this case is $v_i + V[i-1, j-w_i]$

choosing max of (1) & (2) gives

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

with boundary conditions, and out of bounds values, we have

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

This all makes sense only if weights $w_1, \ldots, w_n, W$ are integers. what if they're not? approx. all weights by rational numbers, calc. lcm of all denominators, then convert all weights to units of $\left(\frac{1}{\text{lcm}}\right)$.

Ex. See p. 6 of handout

Exercise

write an algorithm that, given input vectors $v=(v_1,\ldots,v_n)$, $w=(w_1,\ldots,w_n)$ and limit W, fills in the table $V[1\ldots n; 0\ldots W]$ and returns $V[n,W]$. (alternately, return whole table $V[\cdot,\cdot]$)

Exercise. (recursive)

write an algorithm that backtracks through table $V[\cdot,\cdot]$ and returns a list of exactly which objects to steal, or returns $x=(x_1,\ldots,x_n)$.

The Principle of optimality

an opt. problem satisfies the Principle of optimality (also optimal substructures) if an optimal solution to any (non-trivial) instance is a combination of optimal solutions to subinstances.

in our examples:

- coin change:

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

- Discrete Knapsack:

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

showed that problem exhibited optimal substructure.

Defn

a u-v path in a graph $G=(V,E)$ $\begin{cases} u \in V \\ v \in V \end{cases}$ is an alternating seq. of vertices & edges, starting at u, ending at v, in which no vertex is repeated (unless $u=v$). The length of a path is the # of edges in the seq.

notation:

$d(u,v)$ = length of a min length u-v path.

$\ell(u,v)$ = length of a max length u-v path.

Problem 1

given u, v, find $d(u,v)$ and determine a u-v path of len. $d(u,v)$

Problem 2

given u, v, find $l(u,v)$ and determine a u-v path of len. $l(u,v)$.

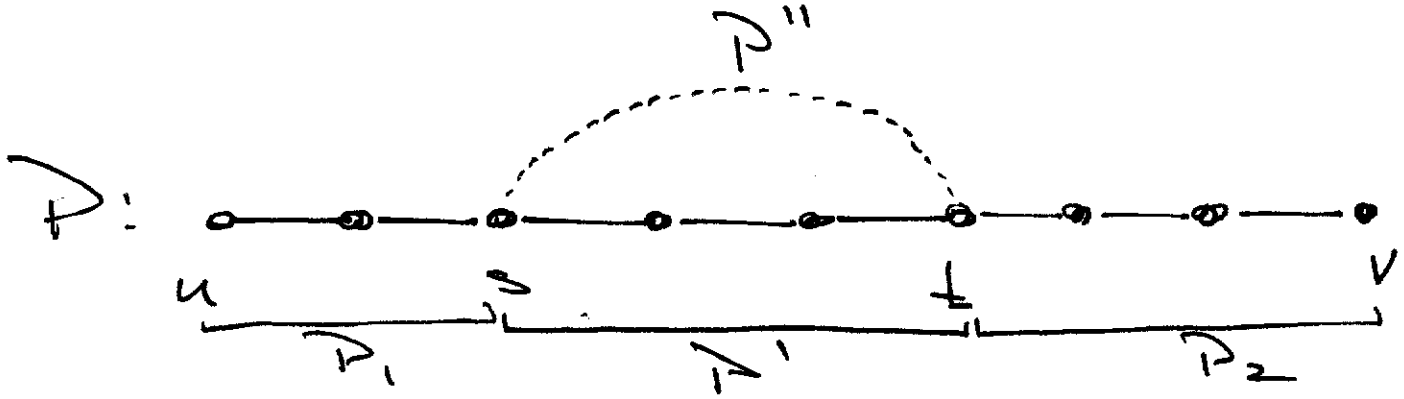
Why are these fundamentally different ?

Thm

Any subsequence of a shortest Path is a shortest Path.

Proof.

Let P be a shortest u-v Path in G, let s and t be two intermediate vertices on P.



let P' be the sub-path of P from s to t. must show P' is a shortest s-t Path. Assume, to get a $\lightning$, that there is an s-t Path in G shorter than P', call it P'', i.e.

$$len(P'') < len(P').$$

Let P_1 be segment of P from u to s, P_2 the segment from t to v. Then

$$len(P) = len(P_1) + len(P') + len(P_2).$$

Let W be walk from u to v obtained by concatinating P_1, P'', P_2.

Then

$$len(W) = len(P_1) + len(P'') + len(P_2)$$

$$< len(P_1) + len(P') + len(P_2)$$

$$= len(P).$$

This contradicts that P is a shortest u-v Path in G. Hence P' is a shortest s-t Path. ∎