CSE 201 2-16-20

Continue coin change Problem . . . .

$c[i,j]$ = min # of coins needed to Pay j units using only coins $\{d_1, \ldots, d_i\}$

for cell (i,j) we have 2 choices!

(1) use no coins of value d_i. The least # of coins is then

$$c[i-1, j]$$

or

(2) use at least 1 coin of value d_i. The least # of coins is then

$$1 + c[i, j-d_i]$$

Thus $C[i,j]$ should be the best option

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

<u>note</u>: • if $i=1$ or $j<d_i$ then one value is outside table $1 \le i \le n$, $0 \le j \le N$. we regard these values as ∞.

• if both $i=1$ and $j<d_i$, then set $C[i,j]=\infty$, indicating that its impossible to pay j units with $\{d_1,\dots,d_i\}$.

<u>Ex.</u> see handout

<u>Ex.</u> " "

<u>Problem</u>: discrete Knapsack

A thief wishes to steal n objects labeled $\{1, 2, \ldots, n\}$. Let

$$v_i = \text{value of obj. } i$$
$$w_i = \text{weight of obj. } i$$

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

<u>Goal</u>: ~~steal~~ max poss. value of goods.

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

<u>Goal</u>. determine a bit string $x = x_1 x_2 \cdots x_n$ so as to

$$\text{maximize}: \sum_{i=1}^{n} x_i v_i$$ ← <u>objective function</u>.

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

Again, there are 2 Problems

- find value of an opt. soln
- find opt. soln itself, i.e. find a particular bit string $x_1 \cdots x_n$ that achieves maximum val.

Let $1 \le i \le n$, $0 \le j \le W$, and

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

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

(1) do not include obj i : max value is $V[i-1, j]$.

or (2) Include obj i : max value is $v_i + V[i-1, j-w_i]$

so we choose the best option!

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

Include boundary & out of bounds values:

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

<u>Ex</u>: see handout

Principle of optimality (optimal substructure)

Defn

an opt. problem satisfies Principle of optimality iff an opt. solution to any (non-trivial) instance is a combination of some of its optimal subinstance solutions.

Ex

- coin change

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

- Discrete Knapsack

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

Let $G = (V, E)$ be a graph.

- a u-v path in G (where $u, v \in V$) is a seq of vertices

$$u = x_0, x_1, x_2, \ldots, x_k = v$$

where $\{x_i, x_{i+1}\} \in E$, such that all vertices are distinct (except poss. $x_0 = x_k$, in which case the path is closed.)

. (length of path) $= k$.

- $d(u,v)$ = length of a shortest u-v path.
- $l(u,v)$ = length " " longest " " .

Problem 1

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

Problem 2

- find $\ell(u,v)$
- find a u-v path of len. $\ell(u,v)$.