CSL 356 Lecture 19 Sept 12

Knapsack problem revisited

Capacity = B

weights $w_1, w_2 \ldots w_n$

profits $p_1 \quad p_2 \quad \ldots \quad p_n$

Max $\sum x_i \cdot p_i$ s.t.

$$\sum x_i w_i \leq B$$

$$x_i \in \{0, 1\}$$

B, w_i's are integral

Define $F_i(y)$ = the maxm profit for the knapsack problem restricted to items $1, 2, \ldots i$ and capacity y

$0 \leq y \leq B$, $1 \leq i \leq n$

Then our objective is to compute

$$F_n(B)$$

$$F_i(y) = \max\left\{ F_{i-1}(y - w_i) + p_i ,\; F_{i-1}(y) \right\}$$

max profit including the i^{th} object

max profit not including i^{th} object

It is a recurrence which implies that if we have "precomputed" the two terms on the RHS, with additional $O(1)$ computation, we can compute $F_i(y)$

$F_{i-1}(y-w_i)$ $\quad$ $F_{i-1}(y)$

i . . . .

base cases

$F_0(y) = 0 \qquad F_i(0) = 0$

$F_i(j)$ is $-\infty$ for $i, j < 0$

$i \backslash y$	0	1	2	3	y		B
0	0	0	0	0	0	0 . . .	0
1	0	.	.	.	.	.	.
2	0	x					
3	0	x					
4	0	x					
	0	.					
i					$F_i(y)$		
$\vdots$							
n	0						

$F_n(B)$

$$F_i(y) = \max \{ F_{i-1}(y - w_i) + p_i, \; F_{i-1}(y) \}$$

We want to compute the entries in an order such that the two required terms precede the current term

What is the time complexity of this process of filling up the table?

$$O(n \cdot B)$$

total # entries in table × $O(1)$ for filling each entry

Input size $= N$: description of n items $p_1\ p_2 \ldots p_n \quad w_1\ w_2 \ldots w_n$

B

$|p|, |w| \le b$ (bits)

$$N = 2b \cdot n + \log_2 B$$

$B \le n^3$, suppose then polynomial in N

$b = 3$

$\log_2 B = n$

$$N = 6n + n = 7n$$

→ Dynamic Programming

→ Pseudo polynomial time algorithm

(it is polynomial in some parameters but not all)