# COL 702 Lecture 16 Sept 21

Given a knapsack with capacity B and n items $x_1, x_2 \ldots x_n$

weights $w_1, w_2 \ldots w_n$

profits $p_1, p_2 \ldots p_n$

Let $F(i, j)$ represent the maximum profit for items $x_1, x_2 \ldots x_i$ $(i \leq n)$ and capacity $j$ $(\leq B)$

(All values of $w_i$, $p_i$ are integral)

Then the original problem is given by $F(n, B)$

$$F(1, j) = \begin{cases} p_1 & \text{if } w_1 \leq j \\ 0 & \text{otherwise} \end{cases}$$

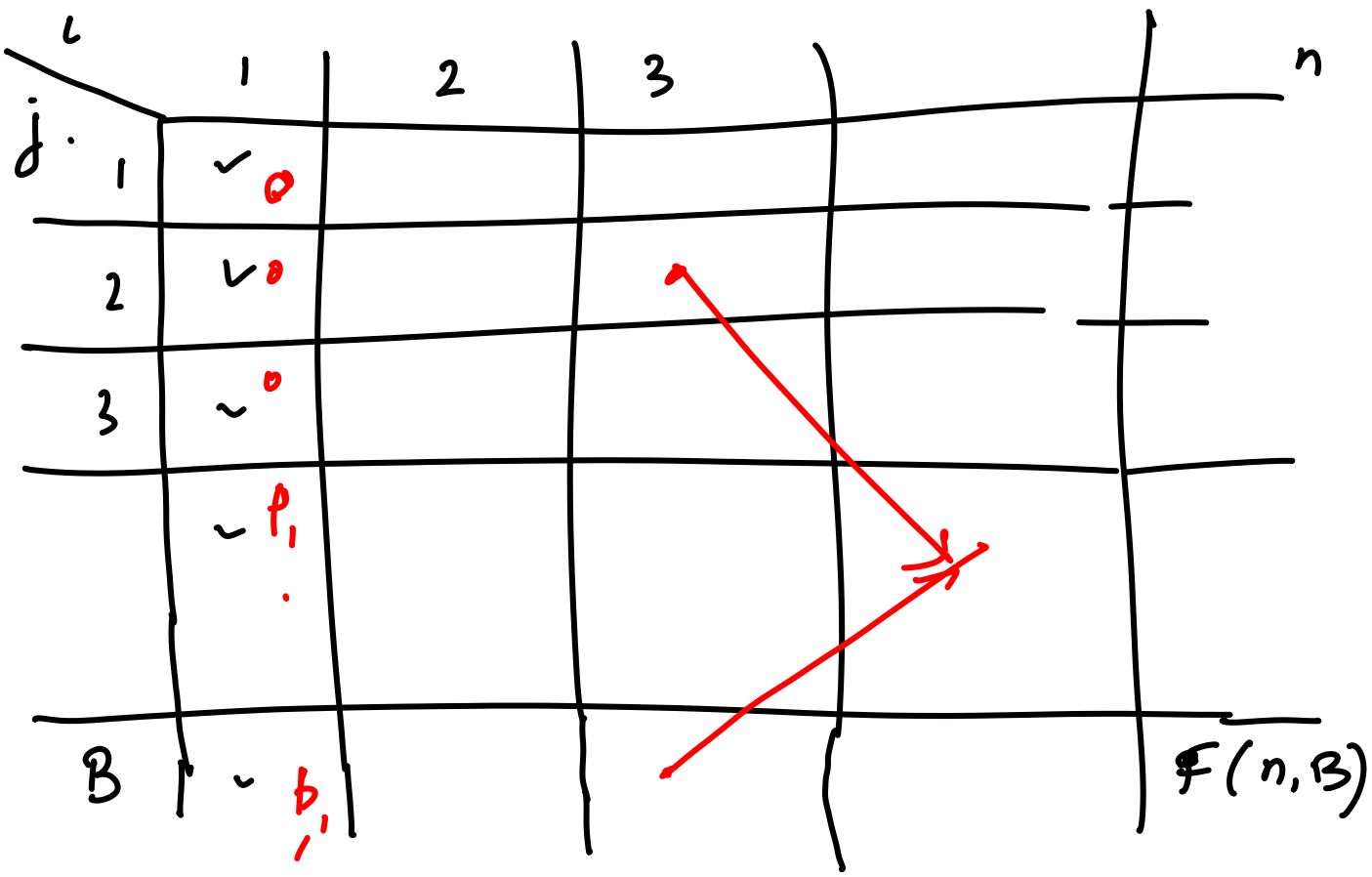
The ordering of elements is arbitrary but fixed in the beginning

$$F(i,j) = \max \left\{ F(i-1, j-w_i) + p_i \;,\; F(i-1,j) \right\}$$

object $x_i$ is included

object $x_i$ not included

$$j - w_i \geq 0$$




How many entries : $n \times B$

To fill each entry we have to look up = 2

Total computation is $O(nB)$

Space ? 2 columns (previous and present)

$O(B)$

Is $nB$ polynomial in $N$?

where $N$ is the size of input "$2n+1$"

$N$ : $2n$ nos representing profits / weights

1 nos representing $B$

$B$ is a no. that requires $\log B$ bits

Suppose $B$ is an $n$ bit no.

Then $N = 3n$ (each profit / wt should also have same length)

$\leq 10n$

Then $nB \leq n \cdot 2^n \sim 2^{N/10}$

for size $N$ we have run in $2^{N/10}$

However if $B$ is bounded by $n^c$

then the running time is polynomial

By scaling down profits, we can solve knapsack problem in $O\left(\frac{n^2}{\varepsilon}\right)$ -time that guarantees at least $(1-\varepsilon)$ OPT profit where

OPT = optimal soln