Cloud Computing

First AC: Costin-Andrei Oncescu, Romania (31:07)
#AC = 16

problem author: Karol Pokorski

Cloud Computing

Easiest possible version

$F_i = 1$, $f_i = 1$

$C_i = 1$, $c_i = 1$

$n = 1$ (one machine)

(consider the most profitable order)

Cloud Computing

Standard version

$F_i = 1, f_i = 1$

$\cancel{C_i = 1, c_i = 1}$

$n = 1$ (one machine)

$O(m \times c_1)$

dp[cores] – the largest profit to have so many cores

Cloud Computing

Double version

$F_i = 1, f_i = 1$

$\cancel{C_i = 1, c_i = 1}$

~~n = 1 (one machine)~~

two knapsacks

$O(n \times (n \times C) + m \times (m \times C))$

Cloud Computing

Double version

$F_i \leq f_i$ ← works too

$\cancel{C_i = 1, c_i = 1}$

~~n = 1 (one machine)~~

two knapsacks

$O(n \times (n \times C) + m \times (m \times C))$

Cloud Computing

One knapsack with modified items, e.g.:

- a task with weight 5 and value 20
- a machine with weight -7 and value -15

We must end with total weight 0 or smaller.

$$O((n + m) \times (n \times C))$$

Cloud Computing

Sort by f_i, F_i decreasingly.

Then just guarantee that the total weight is 0 or smaller **at every moment of time**.

$O((n + m) \times (n \times C))$

Cloud Computing

The alternative knapsack

$V_i = 1, v_i = 1$

dp[cores] → dp[money]

$O((n + m) \times n)$