2025年4月25日 星期五 14:03

Knapsack Problem (背包问题)

Bag: 10 kg

- △1: 3kg 5 — 10
- △2: 4kg 7
- △3: 5kg 8

0-1 Knapsack

Brute Force. n-items. 2^n $\Theta(2^n)$

W: w_i, $i = 1, \ldots, n$

V: v_i, $i = 1, \ldots, n$

$x \in \{0, 1\}$.

C: capacity of the bag

$$\arg\max \sum x_i v_i, \quad \text{s.t. } \sum x_i w_i \le C$$

$$dp(i, w) = \max\begin{cases} dp(i-1, w), \\ dp(i-1, w - w_i) + v_i \end{cases}$$

w: residual capacity

$$dp(i, w) = \max\{dp(i-1, w),\ dp(i-1, w - w_i) + v_i\}$$

$dp(0, w) = 0$.

$dp(i, 0) = 0$

	1	2	3
w_i	3	4	5
v_i	5	7	8

$C = 10$

$C = 10$

$$dp(i, w) = \max\{dp(i-1, w),\ dp(i-1, w - w_i) + v_i\}$$

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	5	5	5	5	5	5	5
2	0	0	0	5	7	7	7	12	12	12	12
3	0	0	0	5	7	8	8	12	13	15	15

$n+1$ — $C+1$

$(n+1)(C+1)$

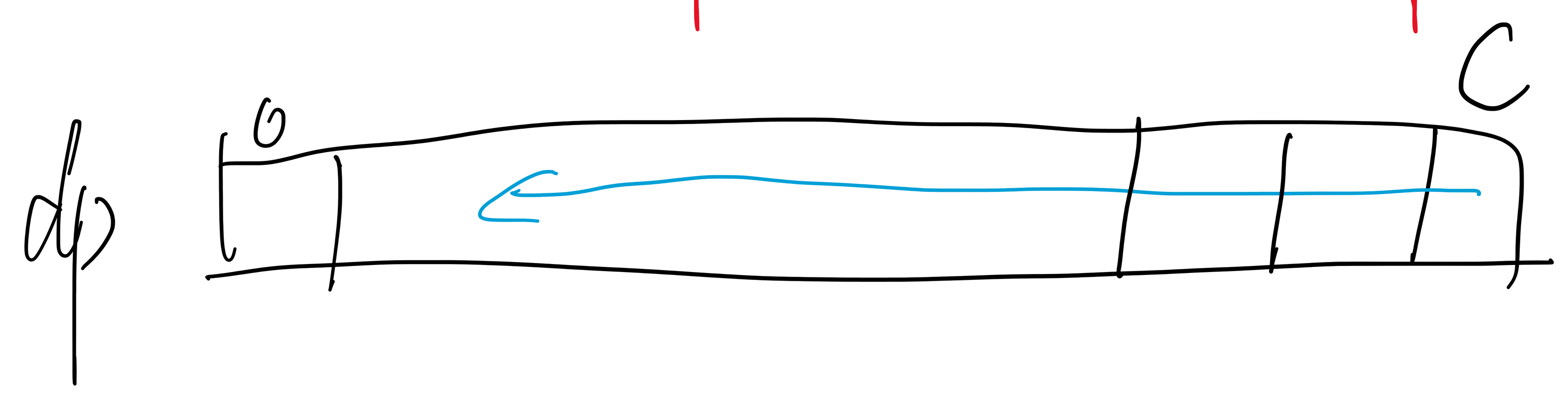
Memory Optimization.

$dp[0 \ldots C] = 0$

for $i \leftarrow 1$ to n

  for $w \leftarrow C$ to w_i

    $dp[w] = \max\{dp[w],\ dp[w - w_i] + v_i\}$



$$dp(i, w) = \max\{dp(i-1, w),\ dp(i-1, w - w_i) + v_i\}$$

$$dp(i, w) = \max\{dp(i-1, w - k_i w_i) + k_i v_i\}, \quad k \in \{0, 1\}$$

Complete Knapsack Problem (完全背包)

$$dp(i, w) = \max\{dp(i-1, w - k_i w_i) + k_i v_i\}, \quad 0 \le k_i \le \frac{w}{w_i}$$

Multi-Knapsak Problem (多重背包)

i: M_i

$$dp(i, w) = \max\{dp(i-1, w - k_i w_i) + k_i v_i\}, \quad 0 \le k_i \le \min\{M_i, \frac{w}{w_i}\}$$

Two Dimension Knapsack (二维背包).

$$dp(i, w, u) = \max\{dp(i-1, w - k_i w_i, u - k_i u_i) + k_i v_i\}, \quad 0 \le k_i \le \min\{M_i, \frac{w}{w_i}, \frac{u}{u_i}\}$$

背包九讲, 2016. 崔添翼.