2021. november 16.

Egy másik Dinamikus Programozás algoritmus a hátizsák feladat megoldására!

① Részproblémák

Érdemes összevetni az előbbivel

OTT : $t_1, t_2, \ldots, t_i$ tárgyakból kihozni a maximális értéket egy j összsúly felső korlát mellett.

ITT: $t_1, t_2, \dots, t_i$ tárgyakból minimális összsúllyal legalább j érték kihozása

Úgy is tekinthetjük ezt, mint egy alternatív hátizsák feladat: a $t_1, t_2, \dots, t_n$ tárgyakból hogyan válasszunk legalább V összértékűt, miközben az összsúly a lehető legkisebb!

Nem nehéz kapcsolatot találni a két hátizsák feladat között, de előbb lássuk az utóbbi DP megoldását!

1. Részproblémák

Minden $1 \leq i \leq n$ és $1 \leq j \leq v_1 + v_2 + \ldots + v_i$ esetén keressük a $t_1, t_2, \ldots, t_i$ tárggyal egy olyan részhalmazát, amelyben az összérték legalább j, és az összsúly minimális

$\rightarrow R[i,j]$

2 Optimális részstruktúra tulajdonság

Legyen O egy optimális megoldása $R[i,j]$-nek. Két eset van:

Ⓐ $t_i \in O$

Ⓑ $t_i \notin O$

Ⓐ Ekkor $O \setminus \{t_i\}$ optimális megoldása az $R[i-1, \boxed{j - v_i}]$ feladatnak

$\max\{0, j - v_i\}$

Ⓑ Ekkor O optimális megoldása az $R[i-1, j]$ feladatnak

Bizonyítás: a szokásos CUT & PASTE

$\boxed{3}$ Rekurzió

Legyen $c[i,j]$ az $R[i,j]$ optimális megoldásához a (minimális) összsúly.

Nyilván $c[i,0] = 0 \quad 0 \leq i \leq n$

különben (na itt azért van némi megfondolni való)

Ha $j > v_1 + v_2 + \dots + v_{i-1}$, akkor t_i benne van az optimális megoldásban, így A szerint

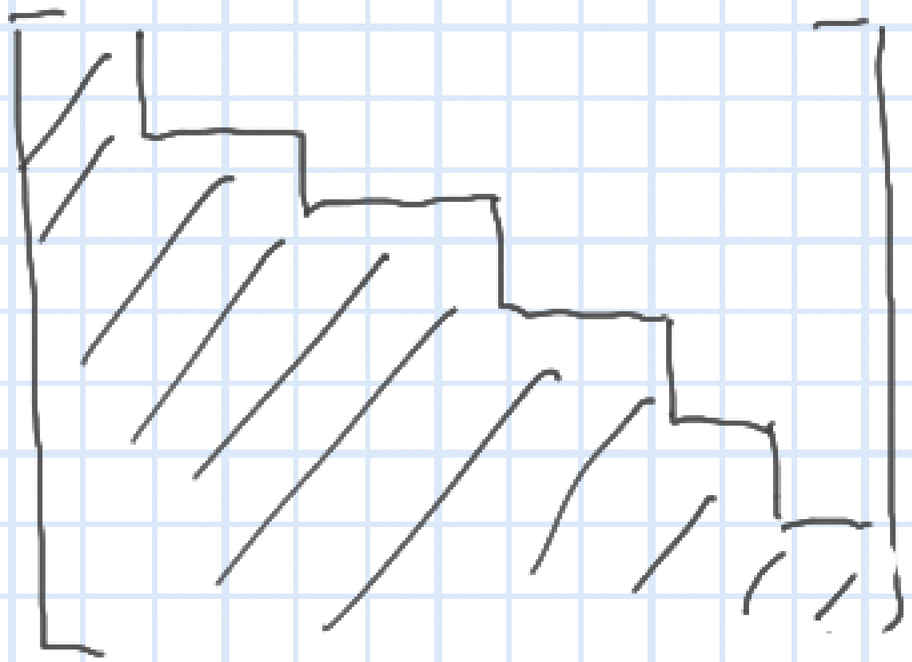
$$c[i,j] = w_i + c[i-1, \max\{0, j - v_i\}]$$

Ha $j \leq v_1 + v_2 + \dots + v_{i-1}$, akkor A és B is előfordulhat optimális megoldásként, vizsgáljuk meg mindkettőt, és válasszuk a kedvezőbbet:

$$c[i,j] = \min \begin{cases} c[i-1,j] & \text{(B)} \\ w_i + c[i-1, \max\{0, j-v_i\}] & \text{(A)} \end{cases}$$

$\boxed{4}$ számolás

A $c[i,j]$ értékek számolása felírt-hető egy $(n+1) \times (v_1 + v_2 + \dots + v_n + 1)$-dimenziós táblázat (parciális) kitöltésével (fentről lefelé, soronként balról jobbra)

$C =$ 

Legyen $v^* = \max\{v_1, v_2, \dots, v_n\}$, ekkor az oszlopok száma $O(nv^*)$, a sorok száma $O(n)$, és egy cellát $O(1)$ költséggel számolunk, így a kitöltés $O(n^2 v^*)$ idő.

Ez megint nem polinomiális!

Polinomiális, ha $v^* = O(n^k)$ valamilyen rögzített k-val.

$\boxed{5}$ optimális megoldás:

mindig feljegyezzük, hogy t_i beválasztása vagy kihagyása volt-e kedvezőbb.

Kapcsolat a két hátizsák feladat között: