АНАЛИЗ НА ЗАДАЧА
АНТИКВАРСТВО

Първа подзадача

Ограничението $l_i = 1$ свежда задачата до класическата задача за раницата, тъй като всяка група предмети всъщност е отделен предмет с дадени стойност и тегло, който можем да изберем без допълнителни разходи. Решаваме я със стандартно динамично $dp[i][j]$ – максималната стойност на подмножество от предмети с номера не по-големи от i и със сумарно тегло j.

Сложност: $O(N \times M)$

Втора подзадача

Тази подзадача е лека модификация на класическата задача за раницата, в която предметите са разделени в групи и не можем да изберем повече от един предмет от всяка група. Тъй като $C = 0$, отново нямаме допълнителни разходи за избирането на предметите. Стейтът на динамичното е подобен: $dp[i][j]$ – максималната стойност на подмножество от предмети, избрани само от първите i групи и със сумарно тегло j. За удобство ще означим с $dp[i]$ максималните стойности, получени за всички възможни сумарни тегла j. Преходът между $dp[i-1]$ и $dp[i]$ трябва да позволява да вземем предвид всички предмети от i-тата група, но най-много един от тях да участва в множеството от избрани предмети за някое j. Разглеждайки предмет със стойност v и тегло w, прилагаме рекурентата формула от предишната подзадача: $dp[i][j] = max(dp[i][j], dp[i-1][j-w] + v)$. Разликата тук е, че я прилагаме многократно и така хем отчитаме предишните предмети в съответната група (чрез $dp[i][j]$), хем не можем да изберем повече от един от тях (защото те не са включени в $dp[i-1][j-w]$).

Сложност: $O(\sum l_i \times M)$

Трета подзадача

В тази подзадача вече избирането на предмет, който не е първи в групата си, изисква допълнителен разход. Очевидно не е възможно да ни потрябват повече от $\sum l_i$ жетона, които и предмети да решим да изберем. Ето защо можем да добавим още едно измерение с големина $\sum l_i$ на динамичната таблица $dp[i][j][t]$, в което да съхварянаме броя на жетоните t, използвани до момента. Накрая, от всяка стойност в динамичното ще извадим $\lfloor \frac{t}{K} \rfloor \times C$, за да получим реалната печалба.

Сложност: $O((\sum l_i)^2 \times M)$

Четвърта подзадача

Можем да направим наблюдението, че не е задължително да пресмятаме цената за купуване на жетоните след изчисляването на динамичната таблица. Това може да се прави по време на самото изчисляване, поддържайки броя на оставащите жетони. Този брой винаги е по-малък от K, понеже няма смисъл да се презапасяваме с жетони, тъй като във всеки един момент може да решим, че ни трябва още една торбичка. Така в рекурентната формула добива вида: $dp[i][j][t] = max(dp[i][j][t], dp[i-1][j-w][(t+t') \bmod K] + v - \lfloor \frac{(t+t')}{K} \rfloor \times C)$, където с t' означаваме броя жетони, необходими за избирането на текущия предмет.

Сложност: $O(\sum l_i \times M \times K)$

Пета подзадача

Размерностите на динамичната таблица, използвана в решението на предишната подзадача, са $N \times M \times K$, което я прави твърде голяма при поставеното ограничение за памет. За 100 точки се налага да се направи оптимизация на паметта, при която се пазят само стойностите за текущото i и за $i - 1$.

Автор: Добрин Башев