АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
МОНЕТА

Означения:

N – брой монети
P – минимална сума на монетите за размяна
C_i – стойност на i-тата монета
D – максимален брой различни видове монети, от които може да запазим поне една
S – сума на монетите, избрани за размяна
T – обща сума от стойностите на всички монети
M – брой различни монети в даденото множество

Подзадача 1

Достатъчно за решаването на тази подзадача е да се направи пълно изчерпване на всички възможности. Сложността е $O(2^N)$.

Подзадача 2

Тази подзадача не се различава особено от стандартната задача за раницата. Реално имаме N монети, и използването на коя да е от тях намалява стойността на D с единица, но ни носи определена стойност към сумата. Трябва да се намери минималният брой монети, които са необходими за образуването на сума поне P. Важно е да се направи наблюдението, че понеже всички монети са с различни номинали, те са най-много $\sqrt{2T}$. Това наблюдение има важна роля в пълното решение на задачата. Сложността на полученото решение е $O(N \times T) \approx O(T \times \sqrt{T})$.

Подзадача 3

Чрез съвсем лека модификация на предишната идея, можем да получим работещо решение и за тази подзадача. Трябва единствено да се отдели по една монета от всеки вид, на която да се постави цена 1, а останалите да бъдат с цена 0. Понеже разглеждаме всички монети една по една, сложността ще бъде $O(N \times T)$.

Подзадача 4

Ще разделим монетите в две множества – C_1 и C_2. В множеството C_1 има по точно една монета от всеки вид, а останалите са в множеството C_2.

Допълнителни означения:
$C_1[i]$ и $C_2[i]$ – стойността на i-тия вид монети в едно от двете множества по ред на нарастване на номиналната стойност
$cnt(x)$ – броят монети със стойност x в множеството C_2

Нека първо да намерим стойността на D. За целта ще използваме лаком алгоритъм, който се основава на факта, че за да използваме най-малък брой от монетите, трябва да изберем първо тези с най-голяма стойност. Самостоятелно той би донесъл на състезателите 20 точки и е следният:

1. Намираме сумата на монетите в множеството C_2;
2. Ако тази сума е не по-малка от P, то отговорът е M, иначе продължаваме към точка 3;
3. Избираме най-голямата от неизбраните до момента монети от множеството C_1 и прибавяме стойността ѝ към сумата;
4. Ако сумата е станала поне P, то D е броят на останалите неизбрани монети в C_1, иначе се връщаме на точка 3.

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА МОНЕТА

Второто нещо, което трябва да направим, е да решим задачата за раницата върху множеството C_1. С $dp_1[i][j]$ ще означим минималния брой монети измежду най-малките i на брой от C_1, които трябва да вземем, за да получим сума j. Рекурентната зависимост е:

$$dp_1[i][j] = min\big(dp_1[i-1][j], dp1[i-1]\big[j - C_1[i]\big] + 1\big).$$

Третото нещо, което трябва да направим, е да решим модифицираната версия на задачата за раницата върху множеството C_2. С $dp_2[i][j]$ ще означим минималния брой монети със стойност $C_2[i]$, които трябва да бъдат използвани, ако се опитаме да изберем монети със стойности измежду $C_2[1], C_2[2], \dots, C_2[i]$ със сума j.

$$dp_2[i][j] = 0, \text{ ако } dp_2[i-1][j] \neq -1$$
$$dp_2[i][j] = dp_2[i]\big[j - C_2[i]\big] + 1, \text{ ако } j \geq C_2[i],\ dp_2[i]\big[j - C_2[i]\big] \neq -1 \text{ и } dp_2[i][j - C_2[i]] < cnt(C_2[i])$$
$$dp_2[i][j] = -1, \text{ ако нито едно от предишните не е вярно}$$

Остана да комбинираме двете части от задачата:

Нека сме открили сума S_1, за която са необходими $M - D$ на брой от монетите в множеството C_1 т.е. $dp_1[M][S_1] = M - D$. Със сигурност от множеството C_1 остават D неизползвани монети, тоест броят на видовете монети, които не са напълно изразходени, ще е поне D.

Търсим минималната възможна сума S_2 (ако съществува), която да се получи от монетите в множеството C_2 и такава, че $S_1 + S_2 \geq P$. Със сигурност за да получим сумата S_2 ще трябва да използваме всички монети от C_2 със стойности, равни на някоя от стойностите на използваните монети от C_1, иначе получаваме противоречие, че D е повече от определеното в началото.

От всички валидни сборове $S_1 + S_2$ избираме най-малкия и възстановяваме едно валидно решение, използвайки двете динамични таблици.

<u>Разсъждения за обема използвана памет:</u>

За да можем да възстановим валидно решение, е необходимо да поддържаме двете динамични таблици с размери $unique(C_1) \times T$ и $unique(C_2) \times T$, където $unique(C_i)$ е броят различни номинали в множеството C_i.

$unique(C_1) \leq \sqrt{2T}$ – в най-лошия случай всички монети са различни

$unique(C_2) \leq \sqrt{T}$ – в най-лошия случай има по две монети от всеки вид

$unique(C_1) + unique(C_2) \leq 2\sqrt{T}$ (това неравенство е по-силно от предходните две, взети заедно)

Когато $T = 120\,000$, двете таблици сумарно имат най-много $2\sqrt{T}$ (~700 реда). Паметта е $700 * 120\,000 * 2B \approx 168$ MB, ако ползваме short, което в случая е необходимост. Също така е препоръчително да преоразмерим таблиците в зависимост от $unique(C_1)$ и $unique(C_2)$, за да не заемаме повече от необходимата памет за изпълнението на програмата.

Автор: Добрин Башев