АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА УРАВНЕНИЕ

Нека да означим с neg сборът на отрицателните числа сред a_1, a_2, … ,a_N (ако такива няма, тогава $neg = 0$). Тогава за всяко i, $1 \le i \le N$, е изпълнено $neg \le a_i \le x_i \le M$, откъдето следва $a_i \le x_i \le M - neg$.

За първата подзадача можем за всяко x_i, $1 \le i \le N$, да обходим числата от a_i до $M - neg$, да проверим дали избраните числа имат сума M и да увеличим отговора с едно, ако сумата на числата е M. По този начин за всяко i, $1 \le i \le N$, алгоритъмът се разклонява в около M посоки, то сложността на този подход е $O(M^N)$. В предложеното решение това е реализирано чрез рекурсия с параметри индексът, който ще се обработва в момента, и сумата до момента. Това решение е реализирано в **equation1.cpp**.

Втората подзадача не се различава съществено от първата. Нека параметрите на рекурсията са idx – индексът, който ще се обработва в момента, и sum – сумата до момента. В процеса на работа отговорът при тези параметри ще ни трябва няколко пъти. За да не го изчисляваме всеки път, прилагаме мемоизация. Това решение е реализирано в **equation2.cpp**.

За третата подзадача е достатъчно за всяко x_{idx}, да обхождаме числата от a_{idx} до $M - neg - sum$, където idx е индексът, който ще се обработва в момента, а sum е сумата до момента. Отново използваме мемоизация. Това решение е реализирано в **equation3.cpp**.

Нека числата $x_1, x_2, \dots, x_N$ представляват едно решение на уравнението. Ще разгледаме изображението f, при което на числата $x_1, x_2, \dots, x_N$ се съпоставят $y_1, y_2, \dots, y_N$, където $y_i = x_i - a_i \ge a_i - a_i = 0$, за всяко i, $1 \le i \le N$. Изображението f е еднозначно определено. Ако сега към y_i прибавим a_i за всяко i, $1 \le i \le N$, ще получим оригиналната редица. Следва, че не възможно след прилагане на f върху две различни редици, да получим една и съща редица (в противен случай след прибавянето на a_i към y_i за всяко i, $1 \le i \le N$, ще получим една и съща редица, което е противоречие). Следователно при прилагането на f върху числата $x_1, x_2, \dots, x_N$, ще получим редица $y_1, y_2, \dots, y_N$, която представлява решение на уравнението:

$$y_1 + y_2 + \cdots + y_N = (x_1 + x_2 + \cdots + x_N) - (a_1 + a_2 + \cdots + a_N) = M - (a_1 + a_2 + \cdots + a_N),$$

където числата y_i са цели и $y_i \ge 0$ за всяко i, $1 \le i \le N$. За да решим задачата за $x_1, x_2, \dots, x_N$ и M, е достатъчно да я решим за $y_1, y_2, \dots, y_N$ и $M - (a_1 + a_2 + \cdots + a_N)$.

За четвъртата подзадача ще използваме метода на динамичното оптимиране. Нека означим $M' = M - (a_1 + a_2 + \cdots + a_N)$, а с $dp[idx][sum]$ означим броя начини да получим сбор sum като използваме числата с индекси от 1 до idx включително. Ако $M' < 0$ задачата няма решение, затова ще считаме $M' \ge 0$. Очевидно, $dp[1][sum] = 1$ за $0 \le sum \le M'$, тъй като y_1 може да приеме всички стойности от 0 до M'. Лесно се вижда, че $dp[idx][sum] = \sum_{k=0}^{sum} dp[idx-1][k]$ за всяко idx, $1 < idx \le N$, което се съобразява след оглед на стойностите, които може да приеме y_{idx}, за да се получи сбор sum. Отговорът на задачата е в $dp[N][M']$. Ако това се реализира чрез пресмятане на всеки стейт отделно, сложността ще бъде $O(N * M^2)$ – реализирано в **equation4.cpp**. Ако се реализира чрез префиксни суми или чрез наблюдението $dp[idx][sum] = dp[idx][sum-1] + dp[idx-1][sum]$, сложността става $O(N *$

M) и това преминава успешно и петата подзадача. Това решение е реализирано в **equation5.cpp**.

За шестата подзадача ще използваме и някои комбинаторни съображения. Нека вземем множеството от всички редици с M' единици и $N-1$ нули. Избираме произволна редица от множеството. Вземаме текущ индекс $i=1$ и $y_1 = y_2 = \cdots = y_N = 0$. Докато срещаме единици, ще увеличаваме y_i с 1, а ако срещнем нула, ще увеличим i с 1. Завършваме с $i=N$ и едно решение на уравнението. Сега, ако вместо всяко y_i запишем y_i на брой единици и след всяко такова „представяне“ от единици, запишем по една нула (освен след $i=N$), ще получим редицата от нули и единици, с която започнахме. Ясно е, че от две различни такива редици от нули и единици не можем да получим две еднакви решения $y_1, y_2, \dots, y_N$ (в противен случай след преобразуването на y_i в y_i на брой единици за всяко i, $1 \le i \le N$, ще получим една и съща редица от нули и единици, което е противоречие). Следва, че всяка редица от M' единици и $N-1$ нули представя по едно решение на уравнението. Броят на тези редици е $\frac{(M'+N-1)!}{(N-1)!(M')!}$. Той трябва да бъде изчислен по модул $10^9 + 7$. Модулът е просто число, което не дели нито числителя, нито знаменателя. За да намерим отговора, трябва да намерим реципрочното число на $(N-1)!$ и на $(M')!$ по дадения модул. Това може да стане чрез използване на Малка Теорема на Ферма (реализирано в **equation6.cpp**) или чрез Разширен алгоритъм на Евклид (реализирано в **equation7.cpp**). И двете реализации са със сложност $O(N+M)$, поради нуждата от последователно изчисляване на факториелите. И в двете реализации изчисляването на реципрочно по модул число се прави за логаритмично време.

Автор: Ясен Пенчев