АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
2-ДЕЛИТЕЛИ

Наивното решение, което проверява всяко от числата в интервала за наличие на желаното свойство (линейно или по-лошо), не носи точки.

За алгоритъм, който търси кои цели положителни числа i могат да се представят като $i = j*(j+1) = j*j+j$, започвайки от $j = 1$, са предвидени около 10% от точките. Идеята е да увеличаваме j с 1, като търсим момента, когато стойността $j*j+j$ настигне или надмине M. Тогава започваме да броим, докато тази стойност надмине N. Очевидно, този алгоритъм ще има принципна сложност в порядъка на $\sqrt{N}$.

Усъвършенстването на този алгоритъм води до важни идеи за решаване на задачата, а именно:

- да се съобрази, че ни интересува именно j в този процес, защото то показва точно броя на числата, които имат желаното свойство, щом започнем броенето от 1;
- ако означим този брой до дадено x с $f(x)$, търсеният отговор е $f(N)-f(M-1)$;
- да се потърсят по-точни ефективни методи за определяне на $f(x)$ за големи x.

Естественият отговор на последното съображение е „двоично търсене“, което свежда сложността до логаритмична и предполага бързо решение в исканите граници. Разбира се, може да се мисли и за константно определяне на $f(x)$. Наистина, лесно се съобразява, че ако определим $t = \lfloor \sqrt{x} \rfloor$, то $f(x) = \begin{cases} t, \text{ако } t(t+1) \leq x \\ t-1, \text{в противен случай} \end{cases}$. Тази идея изглежда много проста, но предполага възможности за коренуване. Коренуването работи относително точно за стандартния тип *long double* в C/C++, ако числата не са много големи. Ако примерите са случайно избрани, решението може да е добро и за входни данни с до 36 цифри. За такова решение са предвидени около 50% от точките. По-нататъшните изчисления предполагат повече програмираме: реализация на някои действия с числа, които имат повече цифри (сравняване, събиране, изваждане, умножение, деление на 2), достатъчни за осъществяване на двоично търсене. Към тях ще трябва да се добави и реализация на алгоритъм за коренуване, ако настояваме съвсем буквално да следваме описаната по-горе идея за намиране на $f(x)$, но спокойно можем да минем и без него.

Автор: Иванка Зангочева-Бакалова

Допълнение: Павлин Пеев