конечное количество транзакций. Это одна из главных особенностей блокчейна.

Все операции проводятся между субъектами напрямую. А осуществляются они за счет того, что все участники подключены к одной сети – Blockchain. Процесс шифрования, известный как хеширование, выполняется большим количеством разных компьютеров, работающих в одной сети. Если в результате их расчетов все они получают одинаковый результат, то блоку присваивается уникальная цифровая сигнатура (подпись). Как только реестр будет обновлён и образован новый блок, он уже больше не может быть изменён. Таким образом подделать его невозможно. К нему можно только добавлять новые записи. Важно учесть то, что реестр обновляется на всех компьютерах в сети одновременно. Таким образом, можно сказать, что без блокчейна не было бы и криптовалют.

УДК 373.5.016:004

Матюшёнок А.А.

МЕТОД ПРЕДВАРИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ ПО ИНФОРМАТИКЕ

МГУ имени А.А. Кулешова, г. Могилёв
Научный руководитель: канд. физ.-мат. наук Марченко И.В.

В данной работе излагается метод предварительных вычислений данных для улучшения асимптоматики решений задач в области структур данных и теории чисел.

Рассмотрим на примерах сравнение метода полного перебора и метода генерации объектов.

Довольно часто в олимпиадной практике встречается следующий тип задач: задана произвольная функция $f(x)$, определенная в целочисленных значениях. Требуется определить количество решений уравнения $f(x) = k$ на отрезке$[a, b]$.

Традиционный метод решения таких задач сводится к последовательному вычислению значений $f(x)$ при $\forall x \in [a, b]$, и если $f(x) = k$, то значение переменной, отвечающей за подсчёт элементов, увеличивается на 1.

Задача «Красивые числа»

Число называется красивым, если сумма его цифр в десятичной системе счисления делится на количество цифр в нём. Требуется найти количество красивых чисел, не превышающих число N ($N \leq 10^5$).

Решение. Пусть $A = \overline{a_1 a_2 .. a_k}$, тогда $G(A) = \sum_{i=1}^{k} a_i$ и

$$f(a) = \begin{cases} 1, & \text{если } G(a) mod\ k = 0, \\ 0, & \text{если } G(a) mod\ k \neq 0. \end{cases}$$

Фрагмент кода, который находит значение *G(x)* и число *k* :

```
s:=0;
k:=0;
while a >0 do
begin
s:= s + a mod 10;
a:= a div 10;
k:= k + 1;
end;
```

При решении задачи возможны следующие ошибки учащихся:

1. Зацикливание из-за бесконечного деления числа 0 на 10, если в условии цикла заменить строгое неравенство *a*>0 на нестрогое *a*>=0.
2. Значения переменных s и k в конце программы будут случайными числами, если не обнулить их в начале.

Итоговая асимптоматика решения: $N * Length(M)$, где N – длина отрезка, Length(M) – длина (количество цифр) максимального числа.

Однако существуют примеры задач, где стандартное решение неприменимо в силу того, что оно работает очень медленно, если длина отрезка, на котором необходимо вести поиск, достаточно велика (больше, чем 10^7), а количество искомых элементов – относительно невелико (меньше, чем 10^5).

Задача «2-3 числа»

Целое положительное число называется 2-3 -числом, если имеет вид $2^x \times 3^y$ *для некоторых неотрицательных чисел x и y, т.е. если среди простых делителей числа имеются только числа 2 и 3. Выведите количество 2-3-чисел на отрезке* $[L, R]$.

Ограничения: $1 \le L \le R \le 10^9$. Время исполнения – 1 секунда.

Решение. Значения функции F(x) явно определено в условии задачи и

$$F(x) = \begin{cases} 1, & \text{если } x = 2^a * 3^b; \ \ a, b \in \mathbb{Z}, \\ 0, & \text{если } x \ne 2^a * 3^b; \ \ a, b \in \mathbb{Z}. \end{cases}$$

В данном случае стандартный алгоритм даёт верный результат, но не укладывается в лимит времени.

Для получения полного решения заметим, что чисел до 10^9, являющимися степенью двойки, всего 29, т.к. 2^{30} = 1073741824. Количество чисел, являющимися степенью тройки – всего 18, т.к. 3^{19} = 1162261467, а любое *2-3-число* есть произведение соответствующих степеней 2 и 3, то есть количество таких чисел не превышает 29 × 19 = 551. Поэтому необходимо сгенерировать два массива степеней, а затем последовательно перемножить каждый элемент с каждым.

После выполнения данных операций у нас имеются все необходимые числа в диапазоне $[1..10^9]$ и задача сводится к тому, чтобы за один проход по массиву определить те, которые лежат во введённом отрезке.

Данный метод имеет сложность 30*20*2 = 1200 операций и программа выполняется за 1200 / $10^8 \approx 0.00001$ сек. Итоговая асимптоматика: $\log_2 N \times \log_3 N$, где N – длина отрезка.