

АНАЛИЗ НА ЗАДАЧА ПРЕМАХВАНЕ

Нека дадено просто число p се среща в разлагането на прости множители на всяко от дадените числа поне q на брой пъти (в частност q може да бъде равно на 0). Тогава най-големият общ делител на числата се дели на p^q . За да го увеличим, можем да премахнем тези от числата, които се делят на p минимален брой пъти, при условие че така няма да премахнем всички числа. Забелязваме обаче, че ако разделим всяко едно число на най-големия общ делител, именно тези числа, които p дели минимален брой пъти, вече няма да се делят на p . По този начин сведохме задачата до вече позната такава: да намерим простото число, което дели максимален брой от новополучените числа.

Нека първо да разгледаме как установяваме дали едно число е просто. Съществуват различни подходи, но един от най-ефективните се нарича решето на Ератостен. Чрез негова модифицирана версия можем да генерираме простите числа от 1 до M със сложност $O(M)$, за разлика от класическата версия, която е със сложност $O(M * \log \log M)$. Основната идея е да изчислим най-малкия прост делител на всяко число. За целта обхождаме числата от 2 до M . Ако за текущото число i досега не сме открили прост делител, това означава, че числото е просто и неговия най-малък прост делител е с самото то. В противен случай числото е съставно и вече сме пресметнали необходимия най-малък делител. И в двата случая разглеждаме всички числа $x = i * p_j$, където p_j приема стойността на всяко просто число по-малко от или равно на най-малкия прост делител на i . Това, което е необходимо да направим, е да отбележим, че най-малкия прост делител на x е именно p_j . Коректността и бързодействието на този алгоритъм се дължат на факта, че всяко едно число x има точно едно единствено такова представяне. Следва реализация на описания алгоритъм.

```
const int M = 10000000;  
  
int e[M + 1];  
vector<int> p;  
  
for (int i = 2; i <= M; ++ i)  
{  
    if (e[i] == 0)  
    {  
        e[i] = i;  
        p.push_back(i);  
    }  
    for (int j=0; j < p.size() && p[j] <= e[i] && i * p[j] <= M; ++ j)  
        e[i * p[j]] = p[j];  
}
```

Предложеното авторово решение залага на стратегията всяко едно число да се разложи на прости множители и накрая за всяко просто число да видим, в представянето на колко от дадените числа участва. Това може да бъде направено ефективно, използвайки информацията, която модифицираната версия на решето на Ератостен предоставя. Така ако имаме x и знаем неговия най-малък прост делител p , отбелязваме, че p дели x , и повтаряме същото с x/p , докато x не получи стойност 1. За допълнителна оптимизация можем да пресметнем x/p в процеса на генериране на простите числа, за да избегнем делението в последствие.