Задача ЧИСЛО

Пояснение към решенията

Във всичките решения се използва функцията за намиране на най-голям общ делител:

```
long long gcd(long long a, long long b)
{
    while (b != 0)
    {
        long long r = a % b;
        a = b;
        b = r;
    }
    return a;
}
```

и функцията за намиране на най-малко общо кратно:

```
long long lcm(long long a, long long b)
{
    return (a / gcd(a, b)) * b;
}
```

В променливите `low` и `hight` се пресмята долната и горната граница за L-цифрените числа `X`, `Y` и `Z`, от които ще се съставя числото `N`:

```
    long long low = 1;
    for (int i = 1; i < L; i++) low *= 10;
    long long high = low * 10 - 1;
    long long shift = low * 10;
```

Променливата `shift` ще се използва като множител при слепването на `X`, `Y` и `Z`, за да се образува `N`:

```
long long N = 1LL * X * shift * shift + 1LL * Y * shift + Z;
```

Решение за 29 т. – файл `number_29p.cpp`

В това решение се използва изчерпващо търсене в троен цикъл за пробягване на всички стойности на променливите `X`, `Y` и `Z`, всяка в диапазона от `high` надолу до `low`. В общото тяло на тройния цикъл се проверява дали `N` удовлетворява изискванията от условието на задачата.

Анализ за броя операции: Броят на L-цифрените числа е $S = 9 \cdot 10^{L-1}$. В най-лошия случай (когато няма решение) циклите правят S^3 стъпки: За $L = 1$ има $9^3 = 729$ стъпки; за $L = 2$ има $90^3 = 729\,000$ стъпки (минават мигновено). За $L = 3$ има 900^3 стъпки – минават само когато отговорът е близо до 999, иначе надвишават времето за работа.

Решение за 68 т. – файл `number_68p.cpp`

Също се използва троен цикъл, но с някои оптимизации, получени от математически наблюдения: `X` и `Y` се делят на `D` и затова се използва стъпка `D` в съответните цикли. Понеже `Y` и `Z` са делители на `M`, правят се проверки `M % Y == 0` и `M % Z == 0` вътре в циклите.

Анализ за броя операции: Проверката `M % Z == 0` се повтаря в най-вътрешния цикъл. При $L = 4$ и стъпка $D = 1$ за всяко от 9000-те X и около 100 валидни делителя Y, вътрешният цикъл обхожда всички 9000 стойности на Z, което прави $9000 \cdot 100 \cdot 9000 \approx 8.1 \cdot 10^9$ операции, което ще надвиши времето за работа.

Решение за 100 т. – файл `number_100p.cpp`

Също се прилага троен цикъл, но се избягват повтарящи се проверки. За целта се използва масив `divisors[9005]` в който предварително се пресмятат в намаляващ ред всички L-цифрени делители на `M`:

```
int cnt = 0;
for (int i = high; i >= low; i--)
if (M % i == 0) {divisors[cnt] = i; cnt++;}
```

Така циклите по `Y` и `Z` обхождат само `cnt` на брой валидни делители (обикновено под 150) и броят операции при $L = 4$ и $D > 21$ пада под милион.

Забележка: В архива е приложена програма `number_opt_100p.cpp`, която решава задачата в по-общ случай от изискванията на условието, а именно когато ограничението $1 \leq D \leq 9\,999$ важи и за $L = 4$.

Емил Келеведжиев