

## АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА СУПЕР ПРОСТИ

Наивният алгоритъм е да се проверява всяко едно число от зададения интервал дали е супер просто, като за него и за числата, които се получават след премахване на най-десните цифри се прави проверка дали са прости по най-елементарен начин – проверява се дали нямат делители от 2 до корен квадратен от съответното число. Такова решение ще получи 30 точки и реализацията му е в **super\_naive.cpp**.

Най-бързият алгоритъм е да се определят кои са простите числа чрез решето на Ератостен и в един масив от тип **bool** с размерност от 1 до  $10^7$  за всяко число от интервала  $[1,b]$  (използвайки го като индекс в масива) да се запише **true**, ако е просто и **false** в противен случай. След това, преминавайки през всички числа от  $a$  до  $b$ , проверката дали съответното число е просто и дали са прости числата, които се получават от него с последователно премахване на най-десните цифри е елементарна – просто ги използваме като индекс, за да проверим дали съответният елемент от подготвения масив има стойност **true**. В един брояч броим намерените суперпрости числа и, ако стойността му в края е равна на 0, извеждаме „NO”. Такова решение ще донесе 100 точки и реализацията му е в **super.cpp**.

Възможни са и различни междинни по скорост решения, при които най-напред се намират простите числа в интервала  $[2,b]$  не чрез решето на Ератостен (например като ги пазим в масив и проверяваме за делители на поредното число само измежду записаните вече прости числа), а след това, ходейки само по простите числа от интервала  $[a,b]$ , да търсим дали числата, които последователно се получават чрез премахване на последната цифра, фигурират в построенния масив с прости числа. Тук скоростта много зависи и от начина на търсене в масива с простите числа – тъй като той е подреден в нарастващ ред би могло да се търси и двоично. Възможни са и комбинации между генериране на масив с прости числа чрез решето на Ератостен и след това търсене в този масив.

Дори наивният алгоритъм, който беше описан в началото, не е толкова бавен, ако разликата  $b-a$  не е голяма.

*Автор: Пламенка Христова*