

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА НАЙ-ГОЛЯМ ОБЩ ДЕЛИТЕЛ

Директният подход е да се проверят всички възможни интервали. За всеки интервал пресмятаме НОД на всички негови елементи и, ако той е по-голям от 1, актуализираме максималната дължина при необходимост. Тъй като има $O(n^2)$ различни интервала, а пресмятането на НОД на един интервал отнема $O(n \log a_i)$ време, общата сложност на този подход е $O(n^3 \log a_i)$. Такова решение е реализирано в `nod_naive.cpp` — 10 точки.

Нека видим как даденото число k ще помогне за решаването на задачата.

В 5 от тестовете k е търсеният НОД. Задачата се свежда до намиране на най-дългата последователност от числа, които се делят на k . Такова решение е реализирано в `nod_25p.cpp` — 25 точки.

В още 4 от тестовете търсеният НОД е $k \pm 2$. Записваме числата в масив и проверяваме дали $k - 2, k - 1, k, k + 1, k + 2$ са възможни стойности на търсения НОД. Избираме дължината на най-дългата последователност (по условие тази максимална дължина е единствена). Трябва да внимаваме за случаите, когато $k - 2 < 2$ или $k - 1 < 2$, тъй като това не са валидни стойности за НОД. Такова решение е реализирано в `nod_45p.cpp` — 45 точки.

За стойности на $n \leq 500000$ можем да използваме предходната идея. По условие възможните стойности на НОД са общо 101, тъй като k се различава с най-много 50. Проверяваме всички възможности и избираме най-добрия отговор. Такова решение е реализирано в `nod_70p.cpp` — 70 точки.

Проблемът на предходното решение е лимитът на паметта. Все пак можем да запишем редицата от числа в масив.

Обработваме числата едно по едно: проверяваме всяка от възможните стойности на НОД (припомняме, че те са 101 на брой). За всяка от тези стойности поддържаеме текуща дължина и максимална дължина — два масива с по 101 елемента.

След това определяме за коя стойност се получава максимална дължина. Има две неща, за които трябва да се внимава:

1. $k - i$ за $i = 1, 2, \dots, 50$ може да стане по-малко от 2;
2. по условие съществува точно една подредица с максимална дължина, за която НОД > 1 , но това не означава, че при описания алгоритъм ще има само един делител d , за който получената дължина е максимална. Трябва да изберем най-голямата стойност на такъв делител.

Такова решение е реализирано в `author.cpp` — 100 точки.

Автор: Димитър Добрев