	На пълното решение	На частичните решения
Тагове	Метод на показалките Решето на Ератостен Канонично разлагане Броене	Обхождане до корен

Анализ

Ако сте чели предишни мои анализи, този може да ви се стори по-различно форматиран. Ще бъдете прави, защото вече сме на LaTeX.

Подзадача №1

Както винаги оставих подзадача с тестовите примери за обратна връзка от системата.

Подзадача №$2, 3, 4$

Подзадачите имат еквивалентни решения на подзадачи №$5, 6, 7$, но се използва наивно обхождане вместо Метода на показалките. Метода на показалките може да се приложи, защото с добавянето на числа в произведение, броят на делителите му нараства.

Идея за решението

За пресмятането на $\tau(n)$, равно на броя делители на число $n = \prod_{i=1}^{k} p_i^{\alpha_i}$ (където всяко p_i е просто), ще използваме формулата $\tau(n) = \prod_{i=1}^{k} \alpha_i + 1$. С count масив поддържаме броя срещания на всяко просто число от 1 до 2×10^6 в произведението на интервала, за който изчисляваме броя делители. Така, за да се реши задачата с Метода на показалките, трябва да се намери начин отчитането на промените в този count масив при:

- Добавянето на число (преместване на дясната показалка).
- Премахването на число (преместване на лявата показалка).

Това може да се направи по различни начини, описани в следващите части на анализа. Тъй като премахването на число е еквивалентно на добавянето, ще обясня само добавянето.

Подзадача №5

Когато числата в редицата са прости, ние директно може да ги добавим в count масива.

Постигната сложност: $O(N)$

Имплементация: `prodPrimes_22p.cpp`

Подзадача №6

Всъщност, при добавянето на число в count масива, ние желаем да го разложим канонично, след което да увеличим count масива за всеки прост делител в него с по 1 за всяко срещане. За тази подзадача се търси стандартното канонично разлагане, в което за да разложим число x, ние обхождаме всеки възможен делител (от 2 до $\sqrt{x}$).

Постигната сложност: $O(\sum_{i=1}^{N} \sqrt{a_i})$

Имплементация: `prod_67p.cpp`

Подзадача №7

Нека чрез Решето на Ератостен за всяко едно число намерим най-малкия му прост делител. Така, с това преизчисление, ние може да разлагаме числата за сложност, равна на броя прости делители, които са до двоичен логаритъм от числото.

Постигната сложност: $O(max(a_i)\ log_2\ log_2\ max(a_i) + \sum_{i=1}^{N} log_2 a_i)$

Имплементация: `prod_100p.cpp`

Автор: Борис Михов