Анализ на задача `divs`

Тагове: комбинаторика, разлагане на множители, решето на Ератостен

Забележка: С $d|N$ определяме, че d е делител на n.

Решение за 20 точки

Тук за всяка заявка можем да обиколим всички числа от 1 до n и да проверим за всяко дали изпълнява условието.

Сложност: $O(qn \log n)$

Реализация: `divs_mitko_20p.cpp`, `divs_kiki_20p.cpp`

Решение за 40 точки

Заради по-голямото ограничение на n не можем да търсим делителите на числата линейно. Известен по-бърз начин се основава на това, че ако $d|n$, то $\frac{n}{d}|n$. По този начин се получава симетрия и ни е достатъчно да търсим само до $\sqrt{n}$.

Сложност: $O(q\sqrt{n} \log n)$

Реализация: `divs_mitko_40p.cpp`, `divs_kiki_40p.cpp`

Решение за 60 точки

Тук се намира основното наблюдение в задачата. Вече не можем да се възползваме от предишните две решения – трябва да се насочим към друг подход. Вместо да се фокусираме върху делителите, нека фокусът ни е върху НОД на числата. Нещо, което може да ни помогне, е че той трябва да бъде точно 1. Кога две числа могат да имат най-голям общ делител 1? Най-правилно и полезно за решението ни можем да отговорим така – ако не споделят общ прост делител. Така ако фиксираме какви прости делители ще има d, то за n/d остава само един вариант – да има тези, които не участват в d, но участват в n.

Формално, ако S е множеството от прости делители на d, а P е множеството от прости делители на n, то простите делители на $\frac{n}{d}$ са множеството $P \setminus S$, където с $\setminus$ се отбелязва разликата на множествата. Задачата се свежда до това да се намерят броя различни множества от прости делители, които може да приеме d. Отговорът на този въпрос е 2^k, където k е броят различни прости делители на n. Най-просто може да си обясним това по следния начин – всеки прост делител може да го включим или да не го включим в каноничния вид на d, което ни дава два варианта. Този избор прилагаме за всеки прост делител на n.

Сложност: $O(q\sqrt{n})$ в най-лош случай, но в общия случай е значително по-бързо.

Реализация: `divs_mitko_60p.cpp`, `divs_kiki_60p.cpp`

Решение за 100 точки

Тук използваме идеята от предишната подзадача, но намираме по-бързо броя на прости делители на число. Това може да направим с модификация на решето на Ератостен. Вместо да маркираме дали дадено число е просто, пазим броя на простите му делители. Както при нормалното реше, обикаляме неговите кратни и за всяко от тях увеличаваме броя. Накрая просто можем да изведем 2 на степен стойността в масива на индекс n. Друг вариант е за всяко число да пазим най-малкия му прост делител и за всяка заявка да го разлагаме, което обаче има сложност $O(\log n)$ на заявка и би било една идея по-бавно.

Сложност: $O(n \log \log n + q)$

Реализация: `divs_mitko_100p.cpp`, `divs_kiki_100p.cpp`

Решение за 80 точки

Възможна е лека модификация на решето на Ератостен, която дава малко по-бавен алгоритъм от този за пълен брой точки, но не се възползва от наблюдението, свързващо броя на простите делители на число и броя на унитарните му делители.

Както в пълното решение, ще ползваме решето, за да изброим делителите на всяко число, но допълнително, на цената на $O(\log n)$ време, още в решето ще определяме дали един делител d на някакво n е унитарен или не – чрез алгоритъма на Евклид ще намерим най-големият му общ делител, заедно с $\frac{n}{d}$.

Сложност: $O(n \log^2 n + q)$

Реализация: `divs_mitko_80p.cpp`, `divs_kiki_80p.cpp`

За любопитните

Съществува и формула за сумата на *унитарните делители* на число. Разбрахме, че има статии, свързани с унитарни делители, след като направихме задачата, но не смятаме, че състезателите са запознати с тях.

Автори: Димитър Шапатов, Кирил Зашев