

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ТРИЪГЪЛНИК

Когато се въведе точка с координати (X_0, Y_0) , търсим най-отделечената от нея с първа координата X_0 , тя със сигурност е най-горната или най-долната от всички точки с координата X_0 . Аналогично за втора координата Y_0 . За да решим тази задача трябва за всеки X_1 , на който има поне една точка (поне 1 точка с координати (X_1, Y) , където Y е произволно число), да можем бързо да разбирате коя е най-високата и най-ниската точка (и да можем да добавяме и премахваме точки). Това може лесно да се имплементира със `set` (в който ще имаме точки – наредени двойки числа), тъй като операциите добавяне в `set`, премахване от `set`, намиране на най-малък/най-голям елемент от `set` са бързи (със сложност до $\log N$).

Остана да доизясним как се имплементира тази идея. Ще имаме вектор от `set`-ове. Всеки ще отговаря на някоя X координата, ще имаме и `map`, който казва на кой индекс във вектора отговаря даден X . Така когато се въведе нова точка я добавяме в съответния `set` (може да се наложи да създадем нов `set` и да го добавим към вектора). Когато махаме точка я махаме от съответния `set` (за X координатата и). Ако се допуска да има над 1 точка на едни и същи координати трябва да се използва `multiset` или точките да имат освен координати (X, Y) и `ID`. Тогава имаме и `map` от координати към списък с `id`-та на тези координати. По този начин от `set`-а ще може да съдържа над 1 точка с едни и същи координати. При заявка за максимален триъгълник намираме минималния и максималния елемент в съответния `set`, за X и Y координатите и. После намираме коя е по-отдалечена умножаваме двете разстояния и делим на 2. Всичко в този параграф се прави аналогично за X координатата и Y координатата.

Забележка: Отговора може да е число до $2 \cdot 10^{18}$ и може да не е цяло (ако двете страни са с нечетна дължина). Тъй като тип `double` няма достатъчно добра точност използваме `long long` за да запазим $2 \cdot S$ и преди извеждане го делим на 2 и после извеждаме “.5“ ако е нечетно.

Вместо `set` може да се използват два `heap`-а, един за най-голямо и един за най-малко число. И отделен масив в който помним кои точки са изтрети и ги махаме от `heap`-овете когато имаме заявка за максимален триъгълник и се натъкнем на точка отбелязана като изтрита (във върха на `heap`-а).

Автор на анализ и решения: Даниел Атанасов