Анализ на задача fishes

Формално, за дадени множество заявки във формата на ромб, трябва да открием колко от въведените точки влизат в тях.

Първа подзадача

Стандартно, задачата включва подзадача, решена чрез пълно изчерпване.

Втора подзадача

Ограничението върху l_i и y_j свежда неравеството $y_j + |a_i - x_j| \leq l_i$ до $|a_i - x_j| \leq 0$, което пък от своя страна е еквивалентно на $a_i = x_j$. Въпросът за две редици колко числа съвпадат вече е даван като задача, макар и тук да е леко по-сложна версия. Все пак можем да решим подзадачата с *std::map* или сортиране.

Трета подзадача

С тези допълнителни ограничения задачата се е появявала като задача E на SEERC 2018.

Можем да си представим заявките като един-единствен ромб, който плъзгаме от ляво на дясно (sweepline идея). Тогава ще забележим, че за всяка рибка, тя е "активна" и се отброява към общия брой хванати рибки на рибарите в един интервал $[y_j - c, y_j + c]$. Така задачата се сведе до "При дадени m интервала, намерете за всяка от n дадени точки от колко интервала се покрива тя?". Това отново е стандартна задача на тема sweepline.

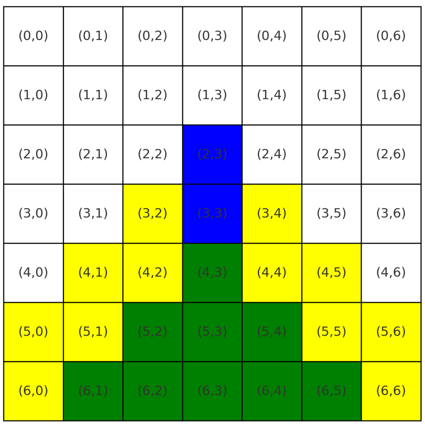
Нека си представим правата Ox за времевата линия и да разглеждаме три вида събития:

- отваряне на интервал;
- затваряне на интервал;
- заявка.

Ще обхождаме събитията от ляво на дясно, а заедно с това ще поддържаме една променлива `balance`, която да ни пази броя отворени интервали. Когато срещнем заявка, `balance` ще ни показва колко интерваала покриват заявката.

Четвърта подзадача

Допълнителните ограничения върху въведението стойности ни позволяват по-леко да обработваме заявките – следващите решения ще се възползват от стандартни техники за B група като коренова декомпозиция, сегментно дърво и дърво на Фенуик. Тук обаче можем да се оправим само с хитри префиксни суми. Нека `pref[x][l]` е броя на рибките, които биха били хванати от рибар на позиция x, който има въдица с дължина l, а `cnt[i][j]` е броят на рибките на позиция (i, j).



Игнорирайте грешното индексиране на клетките.

В сила е следното уравнение, което може да си обясните чрез графиката по-горе.

$$pref[i][j] = cnt[i][j] + cnt[i][j-1] + pref[i-1][j-1] + pref[i+1][j-1] - pref[i][j-2]$$

След като изчислим префиксните суми, отговорът на заяквите е просто $O(1)$ сверка с `pref[x][l]` масивът.

Пета и шеста подзадача

Тук вече трябва състезателят да е наясно с пълното решение (което ще бъде описано в "Седма подзадача"). Подзадачите са, за да възнаградят неоптимални решения (различни видове коренови декомпозиции). Решенията са разписани в *sqrt.cpp* и *sqrtlog.cpp*.

Седма подзадача

Централно за задачата е уравнението $y_j + |a_i - x_j| \leq l_i$. Неприятно в него е абсолютната стойност $|a_i - x_j|$, заради това нека разгледаме как се разтваря тя в зависимост от стойностите на a_i и x_j (взаимното положение на рибар и рибка относно абсцисата).

- $y_j - x_j \leq l_i - a_i$, когато $a_i \geq x_j$;
- $y_j + x_j \leq l_i - a_i$, когато $a_i < x_j$.

Игнорирайки случаят при равенството, двата случая могат да се решат напълно симетрично. Нека решим първия и да оставим втория за упражнение на читателя. Всяка рибка се редуцира до единствена стойност $y_j - x_j$, а всеки рибар до $l_i - a_i$. За пореден път чрез sweepline ще определим колко рибки са "преди" всеки рибар. Единственият по-съществен проблем са големите стойности, които могат да стигнат до 2×10^9 – този проблем ще разрешим чрез coordinate compression. За всяка стойност x, ще намерим намерим такава стойност $f(x)$, така че ако $x < y$ в оригиналните числа, то е вярно, че $f(x) < f(y)$. Имплементацията ще намерите в *author.cpp*.

Най-съществената стъпка в задачата, беше да измислим как да се оправим с $y_j + |a_i - x_j| \leq l_i$. С неудобни функции като $|.|$ е добре да разглеждаме случаи, които опростяват поведението на функцията – за абсолютната стойност това е дали аргументът и е положителен или отрицателен.

Името на задачата е заимствано от песента `Weird Fishes/Arpeggi` на Radiohead.

Автор: Иван Лупов