

Анализ

Първа подгрупа

Тук решението е просто линейно търсене на отговора. Можем да пазим всички риби в multiset и с обхождане да намираме отговора на всяка заявка

*Сложност $O(N*Q)$*

Втора подгрупа

В тази подгрупа интересното е, че имаме заявки само за добавяне и за питане на въпрос. Нека погледнем на задачата от друга гледна точка и си представим различните риби като върхове на граф и, ако съществуват два върха от вида k и $k+1$, то между тях да има ребро. Тогава задачата се свежда до намиране на най-голяма компонента в граф и обединяване на две компоненти. Най-лесният и бърз начин за поддържане на тези две операции е чрез използване на структурата DSU. Така при добавяне на вкус K , ние трябва да проверим дали съществуват върхове $k-1$ и $k+1$, и ако съществуват да построим съответните ребра. За максимално забързване ще използваме компресиране на пътищата и small-to-large обединение.

Сложност $O((N+Q)\alpha(N+Q))$*

Трета подзадача

Тази подзадача много прилича на миналата, обаче вместо да имаме само въпроси и добавяния, имаме въпроси и махания. Важно е да забележим, че задачата не е онлайн, тоест можем да решим заявките в произволен ред и единствено да изведем отговорите спрямо въвеждането на съответните им въпроси. Какво ако разглеждаме заявките наобратно? Вместо да махаме върхове от начален граф, ние ще добавяме върхове към крайния граф. Така свеждаме всички премахвания до добавяния, което е същото като

миналата подгрупа. Първо минаваме през всички заявки и пазим кои върхове за все още в графа. След това построяваме финалния граф и при всяка заявка за махане, ние ще добавяме въпросният връх. Важно е да не забравим да изведем отговорите отзад напред, понеже сме ги отговорили въпросите от последния към първия.

Сложност $O((N+Q)\alpha(N+Q) + Q*\log_2 Q)$*

Четвърта подзадача

Задачата е в почти пълния си вид. Тук вече DSU няма да ни свърши работа (съществува структура наречена „DSU с триене“, но тя е по-сложна за имплементиране и извън конспекта за С група). Трябва да измислим друг подход за решение. Използвайки представянето на вкусовете като граф, можем да забележим, че всяка компонента в този граф е пръчка(всеки връх има 0,1 или 2 съседа). Това ни дава възможност да гледаме на вкусовете като редица от числа. Нека кажем, че ако за даден вкус K съществува поне една риба с такъв вкус, то в масив със индекс K ние ще запишем 1, в противен случай ще запишем 0. И задачата придоби доста по-различен облик. При такава обстановка тя се свежда до намиране на най-дълга редица от последователни единици (между a и b), където на всеки индекс в масива съответства 1 или 0. Това може да бъде решено ефективно чрез сегментно дърво. В сегментното дърво за всеки интервал ще пазим отговора за интервала както и най-дългите префикс и суфикс с последователни единици (можем да си представим, че симулираме алгоритъм „разделяй и владей“ в сегментното дърво). Това ни позволява да обединяваме интервали със константна сложност. Заявките в дървото стават със сложност \log_2 от броя елементи.

Сложност $O((N+Q)\log_2(N+Q))$*

Пета подзадача

С пълните ограничения, решението от 4 подзадача ще изкара ML (няма как да поддържахме сегментно дърво с максимален индекс 10^9). Можем да забележим, че при N начални числа и Q заявки, програмата ще трябва да разглежда не повече от $N+Q$ числа. Това ни дава възможност да използваме „компресия по координатите“. С други думи, ще сортираме числата, участващи във входа, и на всяко ще присвоим фалшива стойност. Така всички числа ще бъдат в интервала от 1 до $N+Q$. С тази компресия има проблем, обаче. Например, ако компресирани вкусовете са в последователност $\{1, 2, 3, 6, 9, 10\}$ и направим запитване за интервал 1-10, когато разполагаме с всички вкусове, то програмата ще изкара отговор 6, когато той всъщност е 3. Проблемът е, че в тази ситуация 3, 6 и 9 ще бъдат разглеждани като последователни числа, каквито те всъщност не са. За да решим този проблем при изпълняване на компресията, когато се появи вкус K , ние ще добавяме сред числата за компресиране както K , така и $K-1$ и $K+1$. Така на всеки реален вкус ще съответстват левият и десният му съсед. Максималният брой вкусове, които ще трябва да се разглеждат, са $(N+Q)*3$. От тук нататък решението е същото като това на четвърта подгрупа.

*Сложност $O((N+Q)*3*\log_2((N+Q)*3))$*

Автори: Преслав Тошев и Калоян Върбанов