

Анализ на задача photo

Тагове: неявен граф, DFS, две показалки

Условието задава, че височините са дробни числа, но това няма нужда да ни притеснява. Единственото важно е, че имаме неизброимо много възможности за различни дробни числа между 1 и 2, затова можем да считаме височините и за произволни цели числа. Също редът на алтерниране няма значение (стига да е един и същ – $\nearrow \searrow \nearrow \dots$ или $\searrow \nearrow \searrow \dots$), така че в някои решения можем да разгледаме и другия ред на алтерниране спрямо условието.

Решение на първа подзадача – 11 точки

Трябва да се възползваме от малкото ограничение за K . По този начин имаме само 2 възможности за височините на двамата ученици – 0, 1 и 1, 0. Нека проверяваме даден подмасив дали е възможен. Един вариант е да проверим, че няма противоречия при някоя от двете възможности за височините. Авторът проверява само една възможност. Приема се, че височината на първия ученик (в подмасива) е 0, а на втория – 1, след което се проверява подмасива за противоречие (дори и първите два номера на ученици в подмасива да са еднакви, то директно ще получим противоречие още в началното сравнение).

Трябва да направим някакво преизчисляване на отговорите, но без да правим нещо по-сложно, сложността ще е $O(N^3)$, което е твърде бавно. Можем да се възползваме от монотонността при фиксиран ляв край на подмасивите, защото с изместване на десния край надясно имаме само повече условия, които трябва да се удовлетворят. Затова е достатъчно за всеки възможен ляв край да намерим $\max$ – колко най-много надясно можем да включим учениците, така че подмасивът да е възможен. Като преизчислим тези стойности в един подмасив, то всяка заявка (x, y) ще отговаряме константно като сравняваме y дали е преди $\max[x]$. Това ще използваме и в последващите решения.

Сложност: $O(N^2 + Q)$.

Решение на втора подзадача – 28 точки (=11+17)

Тук можем да приложим първия вариант от първа подзадача, но трябва да проверим $K!$ възможности за височините на учениците. По-малките ограничения позволяват да направим преизчисляване на всички отговори без наблюдението за монотонност.

Сложност: $O(K!N^2 + Q)$ или $O(K!N + Q)$, ако използваме монотонността.

Решение на трета подзадача – 33 точки

Тази подзадача е за първата стъпка към пълното решение. Малкото заявки трябва да ни подсказват да търсим някакъв по-бавен алгоритъм за намиране на това дали даден подмасив е възможен. По-лесно е да мислим за обратното – кога не е възможно. Тогава с разсъждения можем да видим, че това се случва, ако има някакъв цикъл от отношенията между височините. Най-лесният начин е да разгледаме отношенията между височините на учениците като ребра в граф с върхове самите ученици. Можем да насочим ребрата, така че ребро (u, v) да задава, че височината на върха v е по-голяма от на u . Тогава наистина ако графът има цикъл няма възможно разпределение на височините без противоречия, а ако няма цикъл имаме валидно решение, защото имаме топологична наредба на върховете, спрямо която можем да сложим височините.

Моделирането с граф свежда задачата до определянето дали даден ориентиран граф е цикличен. Така за всяка заявка построяваме графа и го обхождаме със стандартното DFS обхождане за намиране дали има цикъл в ориентиран граф. Този подход всъщност може да реши и предната подзадача със сложност $O(N^2K + Q)$, ако използваме и монотонността (намираме $\max$), така

ще имаме фиксиран ляв край и ще добавяме ребрата едно по едно, като търсим противоречие, проверявайки дали графът е станал цикличен.

Сложност: $O(Q(N + K))$.

Пълно решение – 100 точки

Ще подобрим предното решение като отново преизчисляваме за всеки ляв край най-десния възможен край на подмасив без противоречия. Имаме и още една монотонност – ако $l_1 < l_2$, то и $maxr[l_1] < maxr[l_2]$. Всичко това ни позволява да използваме показалки за намирането на $maxr$. Лявата показалка е за текущия ляв край, а дясната показалка за текущия десен край. Движим дясната показалка, докато графът е ацикличен, а когато стане цикличен, движим лявата показалка. Проверката за цикличност може да е отново с *DFS* обхождане на целия граф, но би могло и да е като само проверяваме дали новото добавено ребро (u, v) участва в цикъл като търсим път от v до u . Последното подобрява константата, но не и сложността, затова N е толкова малко за цялата задача.

Допълнително понеже имаме добавяне и махане на ребра, можем да използваме `unordered_multiset` за списъците на съседите. Можем и да се възползваме от специфичния ред, защото ребрата, които махаме, са или само отзад, или само отпред на даден списък. Така можем да използваме `deque`.

Сложност: $O(N^2 + Q)$.

Някои състезатели може и да са мислили в посока двоично търсене заради монотонността, но всъщност двоичното не ни помага да смятаме по-бързо $maxr$, защото самата проверка за цикличност на графа пак трябва да е линейна. Също някои е възможно да са се подвели с *DSU*, но то не може да се използва, понеже графът е ориентиран. За съжаление засега няма известни алгоритми, които могат да се справят много добре с въпроса за цикличност на ориентиран граф при динамично добавяне на ребра.

Автор: Илиян Йорданов (заета)