Анализ на задача Свързаност

Първа подзадача. За всяка заявка първо ще проверим дали има път и в двете посоки с *DFS*. Ако няма, директно отговаряме с 0. В противен случай пробваме да премахнем всяко едно от ребрата и наново пускаме *DFS* в двете посоки. Ако нито едно от ребрата не прекъсва двата маршрута отговаряме с **M**+1.

Сложност – **O(Q × M × (N+M))**.

Втора подзадача. Тук имаме класическата задача да проверим дали има път и в двете посоки между дадени два върха, която на езика на графите ще рече дали два върха са в една и съща [силно свързана компонента] (накратко ССК). С линейният алгоритъм лесно можем да запишем в коя ССК попада всеки връх и да отговаряме на заявките константно.

Сложност – **O(N + M + Q)**.

Трета подзадача. Оттук нататък ще наричаме силен мост ребро, при чието премахване броят на силно свързаните компоненти нараства. Очевидно, ако някое ребро не е силен мост, неговото премахване няма да окаже влияние върху маршрутите между нито една двойка върхове (ССК няма да се променят). За да отговорим на една заявка, е достатъчно да знаем две неща:

- Дали двата върха са в различни ССК по начало – отговор 0.
- Дали има силен мост, при чието премахване двата върха попадат в различни ССК – отговор съответния номер на реброто
- Нито едно от горните две – отговор **M**+1.

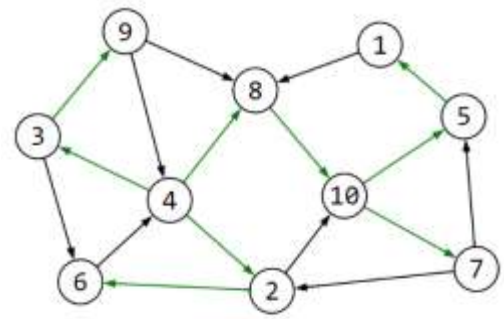
Да разгледаме как бихме могли да отговаряме на втория въпрос без да го проверяваме отделно за всяко обаждане. Не знаем кои ребра могат да са силни мостове, затова ще пробваме да премахнем всяко едно от тях. След всяко премахване намираме ССК и записваме в коя от тях попада всеки връх. Така получаваме таблица с **N** реда и **M** колони – съответно всеки ред показва в кои ССК попада даден връх при премахването на всяко ребро. Сега, за да отговорим на втория въпрос, е достатъчно да намерим първата колона, в която се различават два реда от тази таблица.

Сложност – **O(M × (N+M) + Q × N)**.

Четвърта подзадача. Поради наличието на повече ребра трябва някак да забързаме построяването на таблицата. Да отбележим, че ребрата p → p+1 и p+1 → p за всяко $1 \leq p \leq N-1$ сами по себе си правят целият граф силно свързан. Но, ако те са достатъчни, за да попадат всички върхове в една ССК, възможно ли е някое от останалите ребра да е силен мост? Отговорът е не, защото независимо кое от тях махнем гарантираните $2 \times (N-1)$ ребра в тази подзадача поддържат целият граф силно свързан. Така кандидати за силен мост остават само ребрата между върхове с поредни номера. Единственото изключение са мултиребрата, които очевидно няма как да са силни мостове.

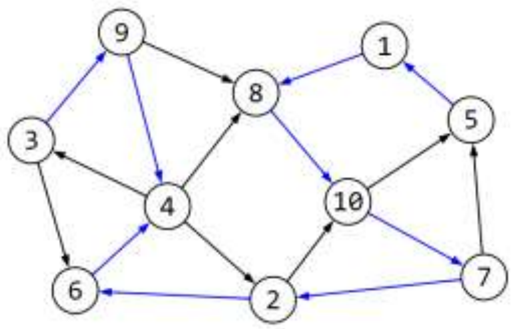
Сложност – **O(N × (N+M) + Q × N)**.

Пета подзадача. Решението на предната подзадача ни кара да се замислим дали наистина силни мостове могат да са много ребра. Тук нямаме ребра, които да гарантират някоя ССК, но нищо не ни пречи да пробваме да намерим такива. Нека сме намерили ССК на оригиналния граф. Ребрата извън тях по дефиниция няма как да са силни мостове.



Едно основно свойство на ССК е, че всеки връх достига до всички останали в нея. Следователно можем да построим покриващо дърво от произволен връх, в което всички ребра сочат от корена към листата.

На илюстрацията в дясно зелените ребра образуват едно примерно такова покриващо дърво с корен връх 4. То може да бъде намерено посредством обикновено *DFS* обхождане, като в зелено оцветяваме ребрата, водещи към непосетен връх.



Другото основно и аналогично свойство на ССК е, че всеки връх е достижим от всички останали. Следователно можем да построим покриващо дърво от произволен връх, в което всички ребра сочат от листата към корена.

На илюстрацията в дясно сините ребра образуват едно примерно такова покриващо дърво с корен отново връх 4. То може да бъде намерено посредством *DFS* по обратните ребра на графа, като в синьо оцветяваме ребрата, водещи към непосетен връх.

Сега нека вземем обединението на тези две покриващи дървета. Оказва се, че те образуват „покриващ силно свързан граф“. Защо това е така? Да вземем да произволни върха u и v. От u можем да стигнем до корена с помощта на второто покриващо дърво и оттам до v с помощта на първото. Аналогично има път и в обратната посока от v до u. Следователно двата върха са в една ССК.

Подобно на миналата подзадача всички ребра, които не участват в тези две дървета няма как да са силни мостове – ССК не се разпада. В най-лошият случай двете дървета нямат общи ребра, което дава и максималния брой кандидати за силни мостове – $2 \times N - 2$.

Сложност – **O(N × (N+M) + Q × N)**.

Шеста подзадача. Остана да оптимизираме и намирането на колона, в която се различават дадени два реда от таблицата. С прекомпют можем да постигнем да запишем отговорите за всеки два реда, но това ще са $\mathbf{N}^3$ стъпки, което е твърде много. В такива моменти стандартно нещо е да направим някакво сортиране и този случай ще сортираме редове лексикографски. Ако означим с $differ[i][j]$ първата колона, в която редове i и j се различават то в сила е: $differ[i-1][j] \leq differ[i][j] \geq differ[i][j+1]$. Ако допуснем противното – $differ[i][j] < differ[i][j+1]$ ще излезне, че редове i и $j+1$ имат по-голям общ префикс от редове i и j, което е в противоречие с лексикографската наредба. Така с показалки можем да преизчислим отговорите много по-бързо.

Сложност – **O(M × (N+M) + N × N × log(N) + Q)**.

Седма подзадача. Комбинираме решенията от пета и шеста подзадача.

Автор: Александър Гатев