Задачата представлява съчетание на няколко стандартни техники и изисква състезателите да бъдат добре запознати с тях, за да я решат. Макар че основната идея е лесно забележима, големият брой въпроси поражда нуждата от ефективно решение, което не е толкова очевидно. Въпреки сравнително високата концептуална сложност на задачата и възможните проблеми по имплементацията ѝ се очаква поне един от участниците да достигне до решение за $100$ точки по време на състезанието.

Формално един въпрос може да бъде зададен по следния начин:

 Даден е свързан граф $G(V,E)$ с $N$ върха и $M$ ребра. За всеки път от $s$ до $f$ намираме реброто с най-голяма стойност. Отговорът на въпроса е минимумът от стойностите на тези ребра.

Едно възможно решение е за всеки въпрос да разгледаме всички пътища между зададените върхове. Това може да се направи чрез обхождане в дълбочина с експоненциална сложност. Тестовете от първата подзадача съдържат малки графи, а в тези от втората подзадача, макар и графите да са доста по-големи, те са дървета и поради тази причина между два върха съществува точно един път. Ето защо първите две подзадачи, при които имаме и само един въпрос, не са проблем за това решение и то получава общо $20$ точки.

Оттук нататък, трябва да направим следното наблюдение. Въпросът за $s$ и $f$ в графа $G(V,E)$ е еквивалентен на въпрос за същите два върха в минималното покриващо дърво на графа $G'(V,E')$. Това свойство, присъщо за минималното покриващо дърво, е в основата на решенията за следващите подзадачи.

Третата подзадача може да се реши, ако намерим минималното покриващо дърво на графа и за всеки въпрос разгледаме пътя между дадените върхове. Не е необходим бърз алгоритъм за нито една от двете стъпки. В предложеното решение за 40 точки се изпозва алгоритъма на Крускал като проверката дали два върха са свързани и намирането на отговора за всеки от въпросите се осъществява с обхождане в ширина. Общата сложност на това решение е $O(M\*log\_{2}M+N\*(M+Q))$.

Последните две подзадачи изискват бърз алгоритъм за намиране на минимално покриващо дърво. В авторовото решение е изпозван алгоритъма на Крускал, но проверката дали два върха са свързани се прави със система от непресичащи се множества. Освен това трябва да се намери и по-бърз начин за отговаряне на въпросите. Нека изберем едно ребро от минималното покриващо дърво, което свързва връх $u$ и връх $v$ и има стойност $w$. С $G''\left(V, E^{''}\right)$ ще означим гората от дървета с върховете от оригиналното дърво и ребрата от минималното покриващо дърво с цена по-малка от $w$. Добавяйки избраното ребро в $G''$, ние обединяваме две различни дървета от гората. Можем да забележим, че реброто с максимална стойност по пътя между произволен връх от едното дърво до произволен връх от другото дърво е именно избраното ребро със стойност $w$. По този начин чрез обхождане на ребрата в нарастващ ред на стойностите им построяваме ново дърво, чиито листа са върховете от $1$ до $N$. За всяко ребро, свързващо върховете $u$ и $v$, създаваме нов връх със стойност тази на реброто и поставяме за негови деца корените на дърветата, в които се намират $u$ и $v$. В новополученото дърво отговорът на въпрос за върхове $s$ и $t$ е стойността на най-близкия общ предшественик на двата върха.

Построяването на новото дърво се извършва с помощта на система от непресичащи се множества. Поради големия брой на въпросите, решения, който намират най-големия общ предшественик на два върха от дървото за повече от $O\left(1\right)$ биха получили $67$ точки.

Решението за $100$ точки използва връзката между задачите за най-близък общ предшественик в дърво и за минимум в интервал на редица. С Ойлерово обхождане на дървото създаваме редица, в която запазваме съответния връх и разстоянието му до корена. Така преобразуваме задачата и най-близкият общ предшественик на два върха е върхът с най-малко разстояние до корена в интервала на редицата от първото срещане на единия връх до първото срещане на другия връх. Вече не е проблем да използваме динамично оптимиране и да построим структура от данни, която позволява бързо намиране на минимум в интервал от редицата. Въпросният алгоритъм има сложност $O(N\*log\_{2}N)$ за прекъмпют и $O(1)$ за всяка заявка. Така общата сложност на решението е $O(N\*log\_{2}N+M\*log\_{2}M+Q)$.

Важно е да уточним и реализацията на системата от непресичащи се множества, тъй като този подход се използва на две места в задачата. Необходимо е състезателите да използват евристика за балансиране на дървото и компресиране на пътя до корена. Ако този момент бъде пропуснат, дори решения, основаващи се на авторовата идея, няма да получат много точки.

*Автор: Добрин Башев*