Задачата се свежда до това да намерим най-близките върхове, в които е поставена забрана и да смятаме сбора от цените по ребрата между тези намерени върхове.

**Бавно решение**

С тривиално обхождане на графа можем да намерим търсените най-близки ограничаващи върхове. Това би станало за линейно време за всяка заявка.

**Оптимално решение**

За достатъчно ефективно решение е необходимо бързо намиране на ограничаващите върхове. Това може да се реализира чрез структурата Heavy-Light Decomposition (HLD). Това е разбиване на множеството от върхове в графа на непресичащи се подмножества от последователни върхове, подредени в списък (без разклонения). При това целта на разбиването е да се гарантира, че пътят между всеки два върха минава логаритмично количество подмножества. При така изградена структура можем да работим с всяко подмножество като линейна последователност, върху която да изградим структури като интервални/индексни дървета, парциални суми и т.н.т.

Как се постига това? Първо, избираме корен на дървото (той е произволен). При така избран корен можем да определим поддърветата на всеки връх. Дефинираме два типа ребра. Ребро между върхове A и B наричаме тежко, ако големината на поддървото на B е поне половината от големината на поддървото на A (реброто има посока от A до B, която се определя от избора на корен и следователно поддървото на B е подмножество на поддървото на А). Останалите ребра наричаме леки. При така дефинираните ребра осъществяваме разбиването по леките ребра, т.е. върхове, свързани с леки ребра, принадлежат на различни подмножества, върхове, свързани с тежки ребра, принадлежат на еднакви подмножества.

Защо това разбиване ни върши работа? Гарантираме си, че подмножествата са във вид на списъци, тъй като от един връх може да излиза най-много едно тежко ребро. Гарантираме си, че пътят между всеки два върха минава през логаритмично количество подмножества, тъй като всяко леко ребро, през което преминем, „премахва“ поне половината от върховете.

В нашия случай е полезно да покрием тези подмножества с интервални дървета. Те ни дават следната информация: кой е последният връх, в който има забрана. По този начин търсенето на най-близкия блокиран връх се разделя на две стъпки:

1. Премини нагоре по HLD, докато не е достигнат LCA (пътят между Иванчо и Сашка минава през техния LCA и няма смисъл да проверяваме върховете над него).
2. Провери дали в текущото интервално дърво има блокиран връх. Ако има, избери последния от тях. Ако няма, мини на стъпка 1.

След като сме намерили блокираните върхове, смятаме разстоянието между тях, като използваме LCA и сумата на ребрата от корена до всеки връх.

Една ключова особеност е, че между един от върховете и върха LCA може да няма блокиран. В този случай се налага да търсим първия блокиран от върха LCA до другия връх. Затова интервалното дърво трябва да ни позволява да намерим както последния блокиран, така и първия блокиран.

Заявка за промяна на забрана се свежда до това да намерим в кое подмножество се намира върха (записваме за всеки връх в кое подмножество се намира) и промяна на стойностите, зависещи от него, в интервалното дърво.

Общо сложността става N\*logN + M\*log2N. Втората степен при логаритъма идва от изпълнението на стъпка 2 за всяка стъпка 1 при заявка от втория тип.