

АНАЛИЗ НА ЗАДАЧА ПЪТУВАНЕ

Графът, представен в задачата, е дърво, тъй като не съдържа цикли. За удобство ще приемем, че коренът на дървото е връх 1. С едно обхождане в дълбочина можем да изчислим големините на поддърветата на всеки връх в графа. Също е необходимо да запомним в кой момент даден връх бива посетен по време на обхождането. Ще се възползваме от факта, че върховете от поддървото на даден връх биват посетени непосредствено след самия него.

Сега нека разгледаме заявка, зададена от върховете u и v . На база на вече събраната информация можем лесно да определим дали v се намира в поддървото на u . Това е изпълнено, ако времето на влизане в u е по-малко от това на v и разликата между тези времена е строго по-малка от големината на поддървото на u (в поддървото на даден връх включваме и самия него).

Ако v се намира извън поддървото на u , вторият връх по пътя между двата върха е прекият родител на u . В противен случай, търсеният връх е някое от децата на връх u . За да установим точно кое от тях е, може да използваме двоично търсене. Търсеният връх се явява последното дете, чието време на влизане в обхождането е по-малко или равно на времето на влизане във връх v . Така получаваме решение със сложност $O(N + M * \log_2 N)$.

Друго възможно решение е за всеки от върховете в графа да запазим заявките, които се отнасят до него. Така чрез едно обхождане в дълбочина можем да намерим отговорите на всички заявки. Ключът е да знаем от всеки връх към кой от неговите съседи сме продължили обхождането. Когато посетим нов връх, ще разгледаме заявките, в които се търси вторият връх по пътя от някакъв връх u до него. Разглеждаме случая, когато u е все още непосетен – тогава очевидно отговорът е прекият родител на u . Иначе, отговорът на заявката е този съсед на u , който сме обходили непосредствено след него. Сложността на това решение е $O(N + M)$, но при така зададените ограничения не се различава съществено по време на изпълнение от предишното.