

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ТЕЛЕФОН 112

Ще представим решението на автора на задачи **no112C** и **no112B** едновременно, тъй като задачата **no112C** е подзадача на **no112B**.

В **no112C** се иска по зададен граф $G(V,E)$, $|V|=n$, $|E|=m$, да се отговаря на множество въпроси от вида „Има ли път от връх u до връх v ?“. Такъв въпрос е еквивалентен на въпроса „Връх u и връх v в една и съща свързана компонента на графа ли са?“. За целта, най-добре е да се извърши едно обхождане на графа (все едно, в ширина или дълбочина) за време $O(m)$, при което да се идентифицират свързаните компоненти и всеки връх да се отбележи с идентификатора на свързаната компонента, на която принадлежи (например началния връх на обхождането на компонентата). Тогаво проверката дали два зададени върха са в една и съща свързана компонента става за константно време и алгоритъмът е със сложност $O(m + Q)$, където Q е броят на обажданията на граждани.

В **no112B**, освен въпросите на гражданите за наличие на пътища между зададени върхове, се добавят и обажданията на пътните служби за почистени пътища. Заявките на гражданите обработваме по същия начин както в задача **no112C**. За бързо обработване на съобщенията на пътните служби е добре да се използва абстрактния тип *разбиване*, популярен още като *find-join-set* или като *find-merge-set*. В зависимост от начина на имплементиране на разбиването, ще получим и различни по бързина алгоритми. Най-добрата известна имплементация на разбиване е с дървета, при което операцията *join* се изпълнява за константно време, а операцията *find*, имплементирана с *повдигане* на дърветата и *балансиране* по височина, се изпълнява за време $O(\log^* n)$, което при малките стойности на n в задачата е равносилно на константа ($\log^* 1000 < 4$). Така получаваме и в този случай алгоритъм със сложност $O(m + Q)$. В предложеното решение не е използвано балансиране на дърветата по височина, тъй като при началното обхождане на графа *find-join-поддърветата* на всички свързани компоненти са с височина 1 и това е напълно достатъчно за получаването на бърз алгоритъм.

Автор: Красимир Манев