## Анализ на решението на задача

# ПЛОВДИВ

За решаване на задачата е нужно знаенето на алгоритъм за максимален поток. Иначе могат да се изкарат много малки точки. Условието задава неориентиран граф, в който трябва да се намерят брой пътища, които не споделят общо ребро. В последните подзадачи има заявки, които добавят нови ребра, което усложнява задачата.

Първата подзадача е за 10 точки. Тя е дадена за хората, които не измислят нищо съществено по задачата и направят някакъв вид пълно изчерпване.

Втората подзадача е за 50 точки. Тя е съществена в задачата, защото за решаването ѝ трябва да се измисли идеята как се намира търсеният максимален брой. Търсенето на максимален отговор може да подскаже подход с максимален поток, който е и правилният. Нека разгледаме дадения граф като потокова мрежа, където всяко неориентирано ребро се дублира, съответно – едното ребро е за едната посока, а другото ребро за другата посока и капацитета на всички ребра е 1. При това дублиране всяко ребро се явява обратно на това, което е в другата посока. Разглеждането на максималния поток от връх номер $\boldsymbol{1}$ до връх номер $\boldsymbol{N}$ дава решение на задачата, защото капацитет 1 на реброто, не позволява да мине път от източника до приемника през повече от 1 ребро, освен това ребрата се разглеждат като потенциални и в двете посоки. Авторът използва реализация на максимален поток с алгоритъма на Диниц, което е достатъчно за решаването на тази подзадача. Очакваната сложност тук е $O(\boldsymbol{MN})$, защото максималният брой пътища без общо ребро са най-много $\boldsymbol{N-1}$ на брой, което се достига в пълния граф, например (понеже не може да използваме едно ребро за повече от един път, а излизащите ребра от връх $\boldsymbol{1}$ са само $\boldsymbol{N-1}$, то няма как да имаме повече на брой от търсените пътища).

Третата подзадача е за 20 точки. Вече имаме заявки, които добавят нови ребра. Очевидно ако всеки път пускаме максимален поток няма да решим оптимално подзадачата. Едно от важните свойства на максималния поток е присъствието на обратни ребра. Те служат за намирането на пътища от източника до приемника, които биха увеличили потока, ако се променят част от старите пътища, които се използват. Това свойство ни позволява да извършваме следното – след всяко новодобавено ребро пускаме максимален поток, но от графа, който е променен, заради минаването на максимален поток досега. Сложността в тази подзадача ще е в най-лошия случай $O(\boldsymbol{QM})$. Следва да се забележи, че $\boldsymbol{M}$ расте с всяка заявка (все пак графът става мултиграф), което дефакто означава, че сложността тук ще е: $O(\boldsymbol{Q^2})$.

Последната подзадача е за 20 точки. Тук ще оптимизираме малко алгоритъма за максимален поток. Ключовите неща са, че ребрата в началото са с остатъчна пропускливост 1, като ако мине поток с големина 1 по ребро, то ще стане с остатъчна пропускливост 0, а обратното ребро ще нарастне на остатъчна пропускливост 2. Това означава, че вместо да пазим графа в списък на наследниците, би било по-удобно да е в матрица на съседство, като пазим за всяко възможно ребро и всяка възможна пропускливост (те са само 0, 1 и 2) броя ребра, които има в графа. Така извършваните $\boldsymbol{DFS}$-и и $\boldsymbol{BFS}$-и при максималния поток няма да работят със сложност $O(\boldsymbol{M})$, а със сложност $O(\boldsymbol{N^2})$. Тук е важно да отбележим, че тестовете са подбрани, така че да не се

извършват много итерации, когато потока след дадена заявка няма да се увеличи, което означава, че сложността за задачата се определя от крайния отговор, който ще се намери, т.е. ще е: $O(\boldsymbol{Q} + 4000 * \boldsymbol{N^2})$.

Задачата може допълнително да се усложни, като се иска проверката дали ще се увеличи потока да се прави с по-добра сложност от $\boldsymbol{N^2}$, защото в най-лошия случай предното решение ще работи със сложност $O(\boldsymbol{QN^2})$. Този проблем е еквивалентен на това: имаме $\boldsymbol{DAG}$, произведен след извършване на алгоритъма за намиране на максмиален поток, заради което той има корен (върхът с номер $\boldsymbol{N}$) и връх, в който свършват най-дългите пътища (върха с номер $\boldsymbol{1}$), имаме заявки за добавяне на неориентирани ребра и за въпроси дали има път от връх $\boldsymbol{1}$ до връх $\boldsymbol{N}$. Понеже решението тук, не е чак толкова просто (за любопитните – възможно е да се реши със амортизирана сложност $O(\boldsymbol{N})$ за ъпдейт заявка (и константна сложност за въпрос) и $O(\boldsymbol{N^2})$ предпроцесване) нямаше смисъл да се има предвид и това за реализация.

*Автор: Илиян Йорданов*