

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА Like

Задачата Like изисква от състезателите както теоретични познания, така и известно количество досетливост за да измислят ефективно решение. Абстракцията на задачата може да се сведе до: „Даден е ненасочен граф. Насочете го по такъв начин, че разликата на броя на изходящите и входящите ребра във всеки връх да се различава с най-много 1“.

За най-опитните състезатели подобна задача би ги насочила към потокови алгоритми (flow), което е напълно грешно (или поне ненужно) в случая. Понякога е недостатък да знаеш твърде много. Всъщност решението е значително по-лесно от това. Малкото теория, която трябва да знаем, е свързана с това какво е Ойлеров цикъл и какви са изискванията за да има такъв. Наистина – ако графът е Ойлеров, то след прилагане на алгоритъма за обхождане (насочвайки всяко от ребрата по пътя) бихме постигнали граф, в който броят входящи ребра е равен на броя изходящи за всеки връх. Графът, за съжаление не е такъв, но защо пък да не го направим? Условието за намиране на Ойлеров цикъл в ненасочен граф е броят инцидентни ребра (който се нарича „степен“ на върха) за всеки връх да е четен. Е, в началния граф ще имаме някакъв брой върхове с четена степен, и други върхове, които са с нечетна. Интересното тук е, че броят на върховете с нечетен брой инцидентни ребра е четен (задача по дискретна математика: опитайте се да докажете защо е така). Какво става ако свържем по двойки тези „нечетни“ върхове с фиктивно ребро между тях? Така след добавянето на фиктивните ребра всеки връх ще е с четна степен, следователно новият граф ще е Ойлеров. Вече казахме, че в него можем да насочим ребрата по такъв начин, че дисбалансът между входящата и изходящата степен на всеки връх (в насочения граф) да е нула. Тъй като преди да си свършим работата прибавихме някакви фиктивни ребра, не е лошо да ги махнем. Какво се променя? Към всеки връх в графа сме добавили по най-много едно фиктивно ребро (за да направим степента му четна). След насочването чрез Ойлеровите цикли това ребро е станало или входящо или изходящо за съответния връх. Ако го премахнем намаляме или входящата му степен или изходящата с 1. Това обаче продължава да изпълнява изискването на задачата! Така алгоритъма за решаване може да бъде обобщен в следните точки:

1. Четем първоначалния граф
2. Разделяме върховете с нечетна степен по двойки и ги свързваме с фиктивни (ненасочени) ребра
3. Ползваме някой от алгоритмите за Ойлеров цикъл за да насочим графа за всяка негова свързана компонента.
4. Премахаме фиктивните ребра

Каква е сложността на този алгоритъм? Всяка от петте точки може да бъде изпълнена линейно по броя ребра плюс броя върхове в графа, тоест $O(N + M)$. Това е и сложността на цялата задача.

В тестовите са дадени няколко, в които входния граф е Ойлеров, така че състезатели, които са се сетили за това свойство на графа, но не са се сетили как да го сведат до такъв, да могат да получат известен брой точки. Освен това при малко на брой ребра (до 20, да кажем) можем да ползваме brute force и да тестваме всяко възможно насочване на ребрата.

Автор: Александър Георгиев