Streets

(Решение)

Както по-опитните състезатели сигурно са забелязали, задачата е модификация на тази за „Китайския Пощальон" в насочен граф. И докато нормалните варианти изискват min-cost max-flow за насочения и blossom algorithm за ненасочения, ограничението за дисбаланс на графа до 42 предлага възможност за сравнително по-лесно решение.

Нека обясним идеята на решението на задачата за Китайския Пощальон. Ако преминаването по всяка улица ви напомня за Ойлерови пътища сте напълно прави – двете задачи са тясно свързани. Всъщност нашата задача е да сведем дадения ни граф до Ойлеров такъв (в случая на насочен граф имаме изискванията графът да е свързан (изключвайки върховете, които не са асоциирани с нито едно ребро) и всеки връх да има равен брой влизащи и излизащи ребра). Освен това трябва да направим свеждането по такъв начин, че да минимизираме общата дължина на графа след промяната му. Нека си представим граф, в който имаме един връх с N излизащи и N+1 влизащи ребра, и още един, който е с M+1 излизащи и M влизащи. Ако намерим път от първия до втория връх и удвоим всяко ребро по него, то в резултатния ни граф двата върха ще са със съответно N+1 излизащи, N+1 влизащи и M + 1 излизащи и M + 1 влизащи ребра. С други думи ще намалим дисбаланса на графа. От теорията на графите знаем, че ако имаме точно един връх с едно излизащо ребро по-малко, отколкото влизащи, то ще имаме и точно един връх с едно влизащо ребро по-малко, отколкото влизащи. Нещо повече, дисбаланс на по-малко влизащи отколкото излизащи ребра K, то и дисбалансът на по-малко излизащи отколкото влизащи ребра също ще бъде K за всички върхове в графа. Така остава само да намерим такъв оптимален набор от пътища от върховете с по-малко излизащи ребра към тези с по-голям брой излизащи, който ни дава минимална допълнителна дължина ако удвоим ребрата по новодобавените пътища.

Тази задача е широко-известна и се решава с минимален по цена максимален поток. Тъй като този алгоритъм е сравнително сложен ние ще използваме друга техника, която е възможна поради едно от ограниченията на задачата. Само ще отбележим, че същото ограничение разрешава дори най-простата имплементация на алгоритъма също да върви в дадения Time Limit и се предполага, че някои от състезателите биха предпочели нея.

Авторовото решение, вместо min-cost max-flow използва техниката динамично оптимиране за намирането на оптималното множество от пътища. Ако разделим в ляво върховете, с по-малко излизащи отколкото влизащи ребра и в дясно тези с по-малко влизащи, отколкото излизащи, създаваме двуделен граф. В него използвайки неколкократно някой от алгоритмите за най-къс път (примерно алгоритъма на Дейкстра) намираме каква е цената да свържем i-тия връх от лявата страна с j-тия от дясната. До тук решението се припокрива с това, разчитащо на min-cost max-flow. Сега, обаче, вместо да ползваме потоков алгоритъм пускаме пълно изчерпване на комбиниранията, като мемоизираме по битова маска на взетите от дясната страна върхове. Тази информация ни е достатъчна, тъй като броят битове в маската ни казва и до кой връх от лявата страна сме стигнали. Тук състезателите трябва да забележат, че ограничението 42 за дисбаланс на ребрата включва както върховете с повече влизащи, така и тези с повече излизащи. Всъщност правилното число е 21, което и ни разрешава да ползваме битовата маска. След приключване на алгоритъма знаем не само каква допълнителна цена ще добави разширяването на графа за да стане Ойлеров, ами и кои ребра да удвоим.

След като сторим това остава да пуснем доста простия алгоритъм на Ойлер за намиране на път, минаващ по всяко от ребрата точно веднъж и връщащ се в началния, с което и задачата бива решена.

В тестовете на задачата са включени доста частни случаи, които състезателите не бива да пренебрегват. Участник, който не е измислил истинското решение може да хване близо 30 точки само справяйки се с частните случаи – когато графът е само с един връх, когато в графа няма ребра, когато има ребро, което не е в силно-свързаната компонента на връх 0 и т.н. Този похват, съчетан с написване на лесния алгоритъм на Ойлер, който по условие е казано, че ще бъде възнаграден с 30% от точките, би дал на състезателя около 60 точки – немалко точки за човек, който не е написал най-сложната част от задачата. Това е направено с цел да се научат учениците да пишат по дадена задача даже да не знаят алгоритъма, който я решава. Това е хубава практика на състезания, провеждани по типа на IOI.

Сложността на представеното решение би могла да се представи като сума на алгоритмите, от които е съставено то и е от порядъка на:

- $O(N^2)$ за проста реализация на силно-свързани компоненти
- $O(N^2 * \log N)$ за дейкстра от всеки връх (може да се направи и по-умно)
- $O(21 * 2^{21})$ = const за динамичното оптимиране
- $O(N + M)$ за намирането на Ойлеровия път

Интересното в случая е, че общата сложност се доминира от третата част и е $O(21 * 2^{21})$. Което на теория е константа и трябва да се игнорира. При дадените ограничения, обаче, това е частта на алгоритъма, която най-много се бави.

За тези, които са писали истинското решение сложността е от порядъка на $O(N^2)$ и би решила оригиналната задача за китайския пощальон със сложност $O(N^3)$ при добра реализация на максимален поток с минимална цена или .

Автор: Александър Георгиев