ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА

София 24-26 ноември 2023

Група В 9-10 клас

Анализ на задача Пътуване

Задачата накратко е: Дадено ни е ориентиран претеглен граф. В някои върхове има винетки, с които след това да можем да прескачаме ребра. Какъв е пътят с минимално тегло и най-много $2 \times M$ ребра от 1 до **N**.

Първа подзадача. Използваме пълно изчерпване с параметри brute(на кой връх сме, изминато разстояние, изминати ребра). Ако изминатите ребра надминат 9, можем да спрем, защото всички ребра са положителни и няма смисъл да повтаряме връх.

Сложност – $O(N^N)$.

Втора подзадача. Тук имаме стандартната задача да намерим най-краткия път от от 1 до **N**, като теглата са само положителни. За целта използваме алгоритъм на Дейкстра.

Сложност – $O((N + M) \times \log(M))$.

Трета подзадача. В графови задачи, в които трябва да се съобразяваме с няколко неща едновременно (както тук с дължината на пътя, но и винетките, с които можем да пропускаме ребра), е най-удобно да направим разширение на графа. Нека построим нов граф, като във всеки връх се съдържат две числа {номер на оригиналния връх, брой събрани винетки}. Съответно, ако имаме ребро между върхове a и b на цена t, то в новия граф ще имаме ребрата:

(a,i) -> (b,i+c[b]) на цена t за всяко $0 \le i \le N - c[b]$

(a,i) -> (b,i-1+c[b]) на цена 0 за всяко $1 \le i \le N - c[b] + 1$

Където c[b]=1, ако във връх b се продават винетки и c[b]=0 в противен случай. В новия граф ще имаме N^2 върха и $N \times M$ ребра. В него търсим най-краткия път от (1,0) до (N,i) за някое $0 \leq i \leq N$. Понеже теглата са положителни отново можем да използваме алгоритъм на Дейкстра.

Сложност – **O((N × M) × log(N × M)).**

Коментар: Понеже маршрутът се състои от не повече от 2×M ребра, можем да докажем, че в нито един момент не биха ни трябвали повече от M винетки, но в случая не ни е позволено да събираме повече от N наведнъж.

Четвърта подзадача. Тук имаме друга стандартна задача – да намерим най-краткия път от 1 до N, но и при наличието на отрицателни тегла. Дори да има отрицателен цикъл няма проблем, тъй като търсим път с не повече от 2×M ребра. Затова ще използваме алгоритъм на Белман-форд. Трябва обаче да го модифицираме малко. Негова основна характеристика е, че на i-тата стъпка е открил най-кратките пътища до всички върхове с не повече от i ребра, но е възможно също така да е открил път с дължина над i. Пример за това е следния граф:

1 -> 2 -> 3 -> ... -> N, където всички ребра са с тегла 1

Ако го пуснем от връх 1 и ребрата са подредени отляво надясно, алгоритъмът още на първата стъпка ще открие най-краткия път до N. Това е проблем, защото е възможно след 2×M стъпки да е открил път по-дълъг от 2×M. За да оправим това, първо обхождаме целия списък от ребра и виждаме кои разстояния могат да се подобрят, и чак след това променяме стойностите им. Така е невъзможно да се подобряват разстояние с по повече от едно ребро наведнъж, но и алгоритъма запазва коректността си.

Сложност – **$O(N \times M)$.**

Пета подзадача. Обединяваме идеите от предните две подзадачи. Правим разширение на графа и в него пускаме модифицирания алгоритъм на Белман-Форд.

Сложност – **$O(N^2 \times (N \times M))$.**

Отпечатване на маршрут. Отпечатването на примерен най-кратък път тук не е така тривиално, както е в повечето задачи. Стандартната техника да следим от кой връх е дошъл най-краткия път до даден връх (или иначе казано да строим дърво на най-кратките пътища) не работи, защото е възможно да имаме отрицателни цикли. А това от своя страна значи, че дори след като сме намерили най-краткия път до N, разстоянията до върховете, които сме ползвали за този най-кратък, продължават да се променят и след това.

Затова ще поддържаме „история“ на най-кратките разстояния за всеки връх. Освен дължина на пътя до връх ще пазим и броят ребра по този път. Когато намерим нов най-кратък път, го добавяме в един списък за съответния връх заедно с броя ребра по него. Забележете, че поради принципа на действие на алгоритъма на Белман-Форд, дължините на пътищата във всеки списък постепенно ще намаляват, а броят на ребрата - ще се увеличава.

С този списък лесно можем да намерим от кой връх е дошло най-краткото разстояние, като просто обходим списъка и следим за дължината на пътя в ребра.

Алтернативно решение. Едно друго решение на задачата е да използваме динамично със стейт dp[връх][брой винетки][брой изминати ребра] със същата сложност.

Автор: Александър Гатев