А Л Г О Р И Т М И   В Ъ Р Х У   Г Р А Ф И

ДОМАШНО ПО "ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ" — СУ, ФМИ, 2019 Г.

Имате право да използвате наготово всички алгоритми, изучени на лекции. Всеки алгоритъм, използван наготово, трябва да бъде посочен чрез името си на български език. Ако такъв алгоритъм има няколко реализации, трябва да се уточни използваната реализация. Когато използвате графи, описвайте подробно процеса на моделиране: какво представляват върховете и ребрата на графа. За алгоритми върху графи указвайте използваната алгоритмична схема (ако има такава) — обхождане в ширина или обхождане в дълбочина.

Задача 1. Пешеходец иска да стигне от едно място до друго, но се пази от силното лятно слънце. Затова той се стреми да ходи в сенките на дърветата. Съставете алгоритъм, който за време $O(n^2)$ намира маршрут с най-малка обща дължина на слънчевите участъци. Вход на алгоритъма:
X [1...n], Y [1...n], R [1...n] — центрове и радиуси на сенките на дърветата (смятаме ги за кръгове);
X_start, Y_start — абсцисата и ординатата на началната точка на маршрута;
X_end, Y_end — абсцисата и ординатата на крайната точка на маршрута.
а) Опишете алгоритъма с думи. **(5 точки)**
б) Анализирайте времевата сложност на алгоритъма. **(4 точки)**

Задача 2. В ориентиран граф с неотрицателни тегла на ребрата да се намери най-къс цикъл (ако има такъв) за полиномиално време.
а) Опишете алгоритъма с думи. **(5 точки)**
б) Анализирайте времевата сложност на алгоритъма. **(4 точки)**

Р Е Ш Е Н И Я

Задача 1. Изчисляваме разстоянията между всеки два кръга, вкл. началната и крайната точка (които тълкуваме като кръгове с нулеви радиуси). Този етап изисква време $\Theta(n^2)$. След това търсим най-къс път по ***алгоритъма на Дейкстра.*** Ако не използваме приоритетна опашка, то времето на втория етап също е $\Theta(n^2)$, следователно времето на целия алгоритъм е $\Theta(n^2)$. Ако използваме приоритетна опашка, реализирана чрез пирамида на Фибоначи, тогава времето на втория етап е $\Theta(m + n \log n)$, следователно времето на целия алгоритъм отново е $\Theta(n^2)$. Не бива да използваме реализацията с двоична пирамида, защото тогава времето на втория етап става $\Theta((m + n) \log n)$, което при плътни графи, тоест при $m = \Theta(n^2)$, какъвто е нашият граф, надхвърля зададеното ограничение на времето: $(m + n) \log n = \Theta(n^2 \log n) = \omega(n^2)$.

Задача 2. Тъй като не е даден връх, през който да минава цикълът, разглеждаме всички върхове, тоест търсим най-къс път от всеки връх до всеки връх. Това може да стане за време $\Theta(n^3)$ чрез ***алгоритъма на Флойд—Уоршал,*** където n е броят върхове на графа. После за време $\Theta(n)$ обхождаме диагонала на получената матрица $n \times n$ и избираме най-малкия елемент. Тъй като j-тият елемент върху диагонала на матрицата е дължината на най-късия цикъл, който съдържа j-тия връх на графа, то най-малкият от диагоналните елементи е дължината на най-късия цикъл. Самия цикъл можем да намерим с помощта на указателите към предходния връх на най-къс път, пазени в отделна матрица от тип $n \times n$. Времето за проследяване на най-късия цикъл е $O(n)$. Общото време на целия алгоритъм е полиномиално: $\Theta(n^3) + \Theta(n) + O(n) = \Theta(n^3)$, поради което този алгоритъм носи пълен брой точки.

Все пак това не е най-бързият възможен алгоритъм. Ако от най-къс цикъл премахнем някое ребро, останалата част е най-къс път. Понеже теглата на ребрата са неотрицателни, можем да използваме вариант на ***алгоритъма на Дейкстра.*** Ако искаме да намерим най-къс цикъл през даден връх s, то с помощта на алгоритъма на Дейкстра, пуснат от s, построяваме дървото на най-късите пътища от върха s до всички други върхове на графа. Нека $d(v)$ е дължината на най-късия път от s до v. След като числата $d(v)$ са намерени с алгоритъма на Дейкстра, обхождаме върховете v на графа, от които излиза ребро към s, и изчисляваме $d(v) + w(v, s)$, където $w(v, s)$ е теглото на реброто. Този сбор е дължината на най-късия цикъл, съдържащ реброто от v към s. Минимумът по всички допустими върхове v е дължината на най-късия цикъл през върха s. Самия цикъл намираме, като добавим реброто (v, s) към най-късия път от s до v, пазен в дървото на най-късите пътища. Построяването на дървото на най-късите пътища по алгоритъма на Дейкстра става най-бързо, ако алгоритъмът използва ***приоритетна опашка, реализирана чрез пирамида на Фибоначи.*** Тогава се изразходва време $\Theta(m + n \log n)$, където n е броят на върховете, а m е броят на ребрата на графа. Търсенето на минимума и на самия цикъл изисква време $O(n)$. Ето защо общото време за намиране на най-къс цикъл през даден връх е от порядък $\Theta(m + n \log n) + O(n) = \Theta(m + n \log n)$.

Обаче върхът s не е даден, поради което пускаме описаната процедура от всеки връх на графа и взимаме най-късия от получените цикли; той е най-къс цикъл в графа. Понеже процедурата се пуска n пъти, то общото време е $\Theta(n\, m + n^2 \log n) = O(n^3)$, т.е. този алгоритъм е не по-бавен от първия алгоритъм. Откриването на втория (по-бързия) алгоритъм се възнаграждава с бонус: удвояват се точките за всяко от подусловията на задачата.