КОНТРОЛНО № 3 ПО ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ ЗА СПЕЦИАЛНОСТ "КОМПЮТЪРНИ НАУКИ" — 2. КУРС, 1. ПОТОК (СУ, ФМИ, 15 АПРИЛ 2021 Г.)

Указания:

1) Имената на всички алгоритми и структури от данни да са на български език!
2) Всички изучени алгоритми могат да се използват наготово.
3) Опишете граф, подходящ за зад. 2: какво представляват върховете и ребрата, на какво съответстват посоките и теглата на ребрата (ако има посоки / тегла). Вие сами избирате и описвате структурата на този граф, обаче няма нужда да го построявате: той е зададен като вход на Вашия алгоритъм.
4) За всеки алгоритъм, който извършва обхождане на граф, да се уточнява какво е обхождането — в ширина или в дълбочина (или няма значение). Да се уточнява и броят на обхожданията.
5) Опишете алгоритмите словесно и ги онагледете с подходящи примери.

Не се приемат решения, в които е нарушено някое от горните изисквания!

Задача 1. Даден е свързан неориентиран нетегловен граф G, два върха S и F и едно ребро $\{A\ ,\ B\}$ от G (където A и B също са върхове на G). Щом графът G е свързан, то между върховете S и F има поне един път, а значи и най-къс път (възможно е да има няколко най-къси пътя между S и F). Съставете алгоритъм с линейна времева сложност в най-лошия случай, който по дадени G, S, F, A и B разпознава дали поне един от всички най-къси пътища между S и F съдържа реброто $\{A\ ,\ B\}$, и ако това е така, отпечатва този най-къс път.

Упътване: Използвайте принципа на Белман: всеки участък от най-къс път също е най-къс път.

Задача 2 (компресиране на данни). Дадени са n картини $k_1\ ,\ k_2\ ,\ k_3\ ,\ \ldots\ ,\ k_n$ с еднаква дължина и ширина и с еднакъв брой пиксели P, който е голямо число. За всеки две картини са дадени пикселите, по които се различават; броят им е много по-малък от P. Тоест картините си приличат (например представляват последователни кадри от филм). Картините трябва да бъдат записани във файл с най-малък обем, тоест трябва да се компресират без загуба на информация. Ако например запишем три от картините изцяло (без да ги компресираме), ще се получи файл с големина $3P$. Но ако запишем изцяло само една картина и разликите ѝ с другите две, то ще се получи файл с големина малко над P, т.е. три пъти по-малък (частта над P се дължи на разликите между картините). Предложете възможно най-бърз алгоритъм, който записва картините във файл с най-малък обем. Пресметнете времевата сложност на алгоритъма в явен вид (като функция на n) при най-лоши входни данни. Обосновете подробно избора на модела и алгоритъма.

РЕШЕНИЯ

Задача 1. Решението се основава на принципа, че всяка част от най-къс път на свой ред е най-къс път. Ако някой най-къс път между S и F преминава по реброто $\{A\ ,\ B\}$, то има две възможности — $SABF$ и $SBAF$. В първия случай участъкът SA е най-къс път между S и A, а пък BF е най-къс път между B и F. Във втория случай SB е най-къс път от S до B, а пък AF е най-къс път от A до F.

Алгоритъм:

1) С помощта на ***две обхождания в ширина*** на G намираме най-къси пътища между двойките върхове S и A, S и B, A и F, B и F, S и F. Едното обхождане на графа G е с начален връх S, другото обхождане е с начален връх F.
2) Ако $|SA| + 1 + |BF| = |SF|$, то алгоритъмът връща пътя $SABF$.
3) Ако $|SB| + 1 + |AF| = |SF|$, то алгоритъмът връща пътя $SBAF$.
4) В противен случай никой най-къс път от S до F не съдържа реброто $\{A\ ,\ B\}$.

Всяка от първите три стъпки на алгоритъма има линейна времева сложност, а четвъртата стъпка — константна. Затова общата времева сложност е линейна.

Задача 2. Моделираме входните данни чрез неориентиран тегловен граф G: върхове са картините $k_1\ ,\ k_2\ ,\ k_3\ ,\ \ldots\ ,\ k_n$; ребро има между всеки две картини; теглото на всяко ребро е броят на разликите между картините, които свързва (краищата на реброто).

Най-малък файл (с големина около P) се получава, ако запишем изцяло само една картина, а за всички останали запазим само разликите им от нея или от друга картина. Тоест търсим подграф T, съставен от всичките n върха на дадения граф и ребрата, съответстващи на разликите, които сме избрали да пазим във файла. Тъй като не бива да губим информация, то всяка картина трябва да се получава пряко или косвено от онази, която е записана изцяло. Ето защо T трябва да съдържа път от върха на картината, записана изцяло, до всеки друг връх. Затова T трябва да бъде свързан покриващ подграф на G. Това изискване е симетрично спрямо върховете, т.е. няма значение коя картина ще изберем да запишем във файла изцяло (по условие всички картини имат еднакъв брой пиксели, а именно P). Има значение кои ребра ще включим в T: сборът от теглата им показва с колко обемът на файла е по-голям от P. Затова T трябва да има минимално общо тегло. Ако допуснем, че T съдържа цикъл, премахването на ребро от цикъла не нарушава свързаността на T и не увеличава теглото на T: броят на разликите между картините е неотрицателно число, тоест теглата на всички ребра са неотрицателни. Следователно имаме право да смятаме, че T е свързан граф без цикли, т.е. дърво. Значи трябва да търсим минимално покриващо дърво T на дадения граф G.

За тази цел имаме алгоритъма на Крускал и алгоритъма на Прим—Ярник. Тъй като графът е пълен, то броят на ребрата $m = \Theta(n^2)$. Добре известно е, че времевата сложност на алгоритъма на Крускал е $\Theta(m \log n) = \Theta(n^2 \log n)$, а времевата сложност на алгоритъма на Прим—Ярник е асимптотично равна на $\Theta(m + n \log n) = \Theta(n^2 + n \log n) = \Theta(n^2)$ при най-лоши входни данни. Следователно най-бърз е алгоритъмът на Прим—Ярник с приоритетна опашка, реализирана чрез пирамида на Фибоначи.

СХЕМА ЗА ТОЧКУВАНЕ

Задача 1 носи общо 20 точки, разпределени по следния начин:
— за анализ на задачата и оформяне на идея: 5 точки;
— за уточнението, че графът се обхожда в ширина: 5 точки;
— за пълно описание на алгоритъма: 5 точки.

Задачата може да се реши и с три обхождания на графа — от S, A и B. Такива решения се приемат, защото също имат линейна времева сложност, обаче константният множител пред нейния порядък е по-голям. Предвидени са допълнителни 5 точки — за решаване на задачата с две обхождания на графа.

Задача 2 носи общо 20 точки, разпределени по следния начин:
— за съставяне и обосновка на модела: 5 точки;
— за избор на алгоритъм: 10 т., ако е избран алгоритъмът на Прим—Ярник; 5 т., ако е избран алгоритъмът на Крускал; 0 т., ако е избран друг алгоритъм;
— за анализ и сравнение на времевите сложности: 5 точки.