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

ПРИМЕРНО КОНТРОЛНО № 3 ПО “ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ” — СУ, ФМИ
(ЗА СПЕЦИАЛНОСТ “КОМПЮТЪРНИ НАУКИ”, 1. ПОТОК, ФЕВРУАРИ – ЮНИ 2018 Г.)

Задача 1. Лицето X научава новина и за един час я съобщава на познатите си. След още един час всеки от тях я съобщава на своите познати и т.н. Предложете най-бърз алгоритъм, който по дадена мрежа от познанства и две дадени лица X и Y намира след колко време лицето Y ще чуе новината за първи път. **(25 точки)**

Задача 2. В клетките на електронна таблица са въведени формули. Например, ако в клетката B8 е написана формулата “= A4 + C7 – 2 ∗ D9”, то трябва първо да се пресметнат стойностите на клетките A4, C7 и D9, а след това — на B8. Предложете възможно най-бърз алгоритъм, който да определя в какъв ред да се изчисляват стойностите на клетките. **(25 точки)**

Задача 3. Даден е свързан ориентиран тегловен граф G с положителни тегла на ребрата. Посочен е и връх s на графа G. Предложете най-бърз алгоритъм с вход — графа G и върха s; изход — масив $\text{cnt}[1\ldots n]$, където n е броят на върховете на G, а $\text{cnt}[u]$ е броят на най-късите пътища от s до u за всяко u.
(50 точки)

Всички алгоритми да се опишат словесно, да се пояснят с пример (вкл. чертеж) и да се анализира времевата сложност. Само най-бързи алгоритми носят точки.

РЕШЕНИЯ

Задача 1. Моделираме условието на задачата чрез неориентиран граф: върховете съответстват на хората, а ребрата — на познанствата. В задачата се търси най-късият път от даден връх X до даден връх Y. Графът е нетегловен, затова можем да използваме ***търсене в ширина***, чиято сложност по време е линейна: $\Theta(m+n)$, където n е броят на върховете, а m — на ребрата.

Задача 2. Тълкуваме входните данни като ориентиран граф, чиито върхове са клетките, а ребрата — зависимостите, породени от формулите. По-точно, има ребро от клетката U към клетката V точно когато формулата във V съдържа името на U. Търсим такава номерация на върховете, че всяко ребро да сочи от връх с по-малък номер към връх с по-голям номер. Задачата се решава с помощта на ***топологично сортиране*** — едно от приложенията на схемата търсене в дълбочина. Ако графът съдържа (ориентиран) цикъл, тогава няма подредба на върховете от описания тип. Откриването на цикъл става също чрез ***търсене в дълбочина.*** Двете приложения на търсенето в дълбочина са сходни, затова проверката за цикъл може да се добави към топологичното сортиране.

Времевата сложност на получения алгоритъм е линейна: $\Theta(m+n)$.

Задача 3 решаваме с помощта на алгоритъма на Дейкстра, в който внасяме следните промени. За всеки връх u освен най-късото разстояние от s до u пазим още и броя cnt[u] на най-късите пътища от s до u, намерени до момента. Инициализираме масива cnt[u] с нули; само cnt[s] инициализираме с единица. При всяка релаксация, т.е. при всяко откриване на по-къс път до някой връх u, включително при откриване на първия такъв път, актуализираме не само пътя и дължината му, но и броя cnt[u]: присвояваме му стойност cnt[v], където v е предпоследният връх от новия най-къс път от s до u (последният връх е u). А винаги, когато алгоритъмът намери друг най-къс път от s до u, т.е. такъв, чиято дължина е равна на текущия минимум за u, прибавяме cnt[v] към cnt[u].

Този алгоритъм има същата времева сложност като алгоритъма на Дейкстра, обаче константният множител е по-голям заради допълнителните операции.

ТОЧКУВАНЕ

Задачи 1 и 2 — по 25 точки за всяка, разпределени по следния начин:
— за моделиране на условието чрез граф: 6 точки;
— за избор на алгоритъм: 7 точки;
— за привеждане на пример: 6 точки;
— за анализ на сложността: 6 точки.

Задача 3 носи общо 50 точки, разпределени по следния начин:
— за избор на алгоритъм: 5 точки;
— за инициализацията на масива cnt: 5 точки;
— за инициализацията на cnt[s]: 5 точки;
— за релаксацията на масива cnt: 10 точки;
— за актуализирането на cnt при намиране на друг най-къс път: 10 точки;
— за привеждане на пример: 10 точки;
— за анализ на сложността на алгоритъма: 5 точки.