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", 2. ПОТОК; ФЕВРУАРИ – ЮНИ 2019 Г.)

Забележка: За всяка задача моделирайте входните данни чрез граф. Обяснете какво са върховете и ребрата на графа в конкретната задача. Ако графът е ориентиран или тегловен, обяснете какво означават посоките, съответно теглата на ребрата в конкретната задача. За всяка задача използвайте наготово някой от изучените алгоритми. Уточнете името на използвания алгоритъм, а ако той се реализира чрез някоя от алгоритмичните схеми "търсене в ширина" и "търсене в дълбочина", тогава уточнете и името на схемата. Илюстрирайте решението на всяка задача с подходящо избрани входни данни.

Задача 1. Шпионин разполага с описанието на компютърна мрежа и иска да подслушва съобщенията между два конкретни компютъра s и t. Самите компютри s и t са защитени срещу подслушване, но другите компютри нямат такава защита. Във всеки миг шпионинът може да подслушва само един компютър — произволен компютър, различен от s и t. Шпионинът не знае по кой от възможните пътища възлите s и t ще си разменят съобщения, затова иска да подслушва такъв възел (компютър) v, през който съобщенията между s и t трябва да преминат непременно, тоест независимо от избрания маршрут.

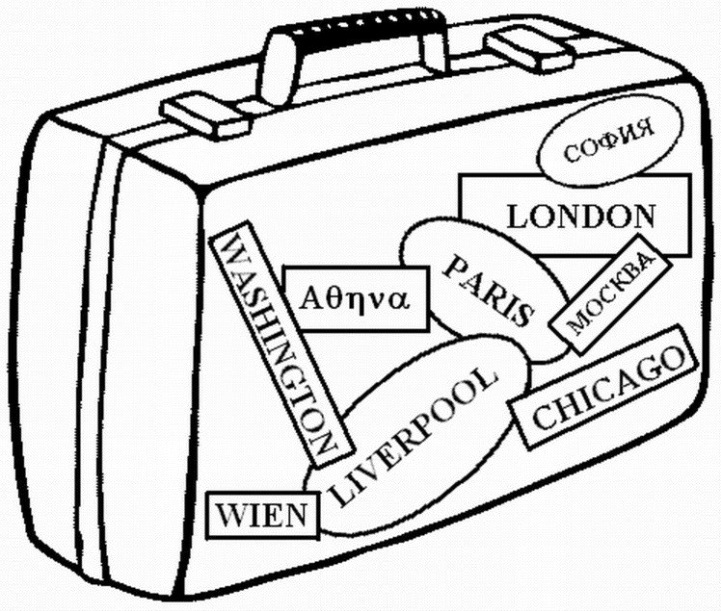
Съставете алгоритъм с линейна времева сложност, който намира възел (компютър) v, подходящ за подслушване (ако изобщо има такъв възел). **(30 точки)**

Задача 2. Разполагаме с данни за мрежа от познанства между множество хора. Един от тях, например Иван, иска да помоли друг, например Петър, за някаква услуга. Те не се познават, затова Иван търси начин да стигне до Петър чрез редица от последователни ходатайства по схемата "познат на моя познат". Всяко ходатайство иска време и прави крайния резултат по-малко надежден, защото всяко следващо лице може да разбере погрешно молбата или да не я предаде по-нататък с необходимата настойчивост.

а) Съставете алгоритъм с линейна времева сложност, който намира редица от най-малък брой ходатаи, свързващи Иван и Петър. **(15 точки)**

б) Да предположим още, че всяко познанство е оценено с цяло положително число, като по-малко число означава по-добро познанство: 1 значи "много добри приятели", 2 — "доста добри приятели", 3 — "обикновени познати", 4 — "здравей-здрасти" и т.н. Съставете възможно най-бърз алгоритъм, който намира редица от ходатаи, свързващи Иван и Петър с най-голяма вероятност за удовлетворяване на молбата на Иван. **(15 точки)**

Задача 3. Върху куфар са сложени лепенки от разни градове. Коя лепенка е сложена първа? (Ако има няколко възможни отговора, достатъчно е да бъде посочен един от тях.)



а) Решете задачата в частния случай, показан на картинката. Отговорът да се обоснове! **(10 точки)**

б) Опишете с думи алгоритъм за общия случай с линейна времева сложност. Илюстрирайте алгоритъма с частния случай, показан на картинката. **(15 точки)**

в) Допълнете алгоритъма така, че да разпознава грешни данни, тоест такива, при които задачата няма решение. Времевата сложност да остане линейна. **(15 точки)**

Р Е Ш Е Н И Я

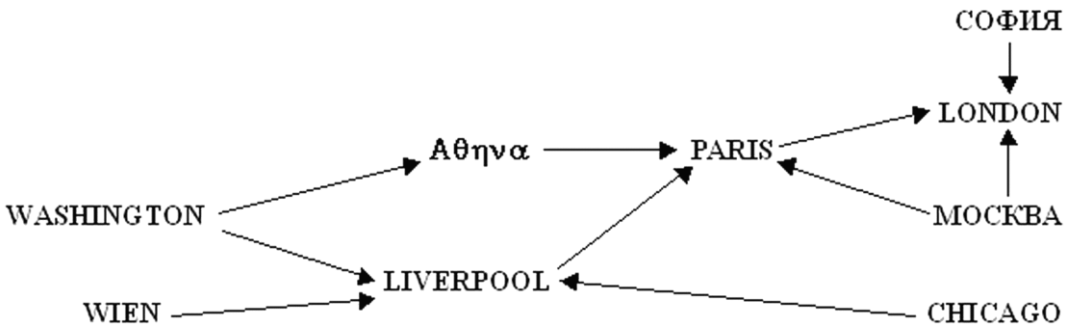
Задача 1. Входът е свързан неориентиран граф: върховете на графа са компютрите, а ребрата са връзките между компютрите. Един компютър v е подходящ за подслушване точно когато е ***срязващ връх*** на графа и лежи на някой от пътищата между s и t (тогава v ще лежи на всички пътища между s и t). Срязващите върхове на графа се намират най-бързо (за линейно време) чрез ***алгоритъма на Шмит*** (а той използва ***търсене в дълбочина***). След като намерим срязващите върхове, търсим един произволен път от s до t (по принцип няма значение как го търсим — в ширина или в дълбочина; но тъй като търсенето в ширина намира най-къс път, то ще приключи по-бързо, отколкото търсенето в дълбочина, въпреки че тази печалба на време не засяга порядъка на времевата сложност в най-лошия случай). Ако върху намерения път от s до t има поне един срязващ връх v, то алгоритъмът връща v като място, подходящо за подслушване. Ако върху пътя от s до t няма срязващ връх, тогава шпионинът няма да може да подслушва със сигурност разговора между s и t.

Задача 2. Входът е неориентиран тегловен граф: върховете на графа съответстват на хората; ребрата съответстват на познанствата; тегла на ребрата са оценките на познанствата. Графът е неориентиран, тъй като предполагаме, че познанството е симетрична релация. (Това предположение не е особено важно: решението по-долу важи и за ориентиран граф.)

а) Търсим свързваща редица с най-малък брой ходатаи, т.е. ***най-къс път***, чиято дължина се мери с броя на ребрата му. Началото и края на пътя са два дадени върха — Иван и Петър съответно. Най-бърз алгоритъм е ***търсенето в ширина.*** То има линейна времева сложност.

б) Пак търсим ***най-къс път*** от един даден връх до друг даден връх (от Иван до Петър), само че сега дължината на пътя е сборът от теглата на ребрата му. В този случай е подходящ ***алгоритъмът на Дейкстра.*** Той е приложим, защото теглата на ребрата са неотрицателни.

Задача 3. Първа може да бъде само лепенка, която не покрива никоя друга. На картинката само лепенката с надпис "**LONDON**" не покрива никоя друга, тоест само тя може да е първа. В общия случай моделираме данните с ориентиран граф: върховете представят лепенките, а ребрата — застъпванията: графът съдържа ребро от върха *x* към върха *y* точно тогава, когато лепенката *x* покрива лепенката *y*. За примера от картинката графът изглежда така:



В общия случай задачата може да се реши с ***топологично сортиране*** на графа, а то използва алгоритмичната схема ***обхождане в дълбочина.*** В подредбата на върховете, получена чрез топологично сортиране, от последния връх не излизат ребра, тоест той съответства на лепенка, която не покрива никоя друга и затова може да бъде първата лепенка, сложена върху куфара. След топологично сортиране графът, изобразен на предишната страница, може да приеме например следния вид (има и други възможни подредби):