

ДДА Семинар 10

Заг. Докажете или опровергнете:

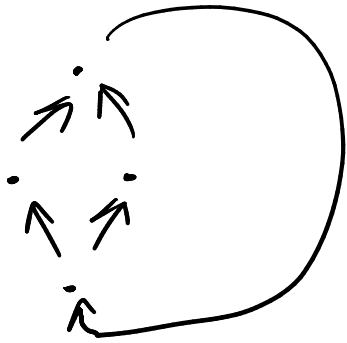
a) $1 \text{ SCC} \Leftrightarrow \exists \text{ HC}$

SCC - strongly connected

HC - Hamiltonian cycle

HP - Hamiltonian path

($\Rightarrow$) Не е вярно:



$\Rightarrow$ За SCC може да има припокриващи се пътища.
За HC не може.

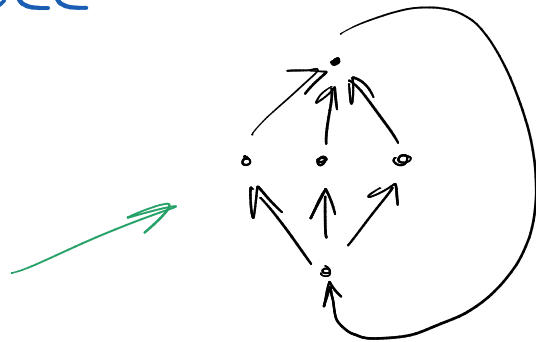
($\Leftarrow$) Да, вярно е:

От всеки връх може да стигнем до всеки друг, използвайки гребена

Те.: $\exists \text{ HC} \Rightarrow 1 \text{ SCC}$

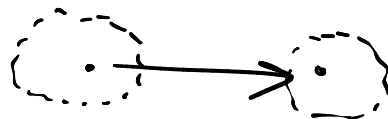
b) $1 \text{ SCC} \Leftrightarrow \exists \text{ HP}$

($\Rightarrow$) Не, както в a)



($\Leftarrow$) Не, контрпример:

Има HP, но 2 SCC



b) $> 1 \text{ SCC} \Leftrightarrow \nexists \text{ HC}$

$A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$

($\Rightarrow$) вярно - от a)

Контрапозиция? Да.

($\Leftarrow$) Не - същия пример като в a)

Ⓟ Заг. В ДАТ има еднств. сорт. $\Leftrightarrow$ ЗНР

Заг Даген е ретни над думи от неизвестна азбука, за които се твърди, че е в лексикографски порядък.

Да се провери дали това е така.

Пр1. abc
acc
baby
bdz
cate

Пр2 abc
acc
bdz
bad
ccc
down

Пр3 7* & ++
a4 % %
a4 & %
& w x x
& ~ a f
~ o o p s

b преди c
a преди b
a преди d

$b < c$
 $a < b$
 $d < a$
 $c < d$

?

→ напр: yadbcz

⇒ Да

$a \rightarrow b \rightarrow c$
↙ ↘
d

⇒ Не, има
цикл

Решение: Искане да проверим дали има азбука, която да се получава от порядъка на думите.

Ще построим граф с върхове за всеки различен символ.

Всяка двойка думи задава ребро за първите различни символи на една и съща позиция.

Следователно, съществува азбука тогава и само тогава когато графът е ациклически.

Графът е ориентиран. Ако няма дъгли е DAG, т.е. може да работим с топологическо сортиране.

Ако има дъгли, ще го открием; ако има цикъл, ще ни я даде.

// Линейна работа $\rightarrow$ топологическо сортиране

• MST - Minimum Spanning Tree

- Kruskal - започва от края от върховете, като по ред добавя връзки Union-Find, обръщайки работата по мено.

$$T(n, m) = \Theta(m \lg n)$$

$$M(n, m) = \Theta(n)$$

• Find - дава ни лидер на компонентата // чрез 2 масива;
Union by rank
По-подробно в лекционните записки

- Prim - започва от произволен връх и алгто добавя работи

Заг

Maximal Shopping

Дизайн и Анализ на Алгоритми, 2011

Time Limit: 0.4s, Memory Limit: 64MiB

Елеонора е кифла. Всеки път, когато отворят нов мол, за нея е малък празник. Тя е толкова въодушевена, че отива да го разгледа още на първия ден от отварянето му.

Явно Ели не е единствената кифла в София, тъй като освен нея там има и огромно количество други хора. Тя е отбелязала нейните **N** любими марки, които имат представителство в новия мол, като е разгледала и какви коридори има в него между двойки от тях и колко хора се движат по всеки от тях. Колкото повече хора – толкова по-трудно тя се придвижва с всичките чанти, които е напазарувала.

Ели е решила да изпълни съставеният от нея план "Maximal Shopping There" и да намери такова подмножество от пътеки, че да може да стигне от всеки магазин до всеки друг (използвайки един или повече коридори) като в същото време сумарно хората по тях са възможно най-малко.

Вход

На първия ред на стандартния вход ще бъдат зададени целите числа **N** и **M** – съответно броят магазини и броят коридори между двойки от тях. Следващите **M** реда ще съдържат по три числа **A_i** **B_i** **P_i**, указващи, че между магазините с индекси **A_i** и **B_i** има коридор, по който минават **P_i** човека. Възможно е да съществуват по повече от един коридор между два магазина, а дори и коридор от магазин до самия себе си (футуристична архитектура). Все пак молът е така конструиран, че винаги да има набор от коридори, които позволяват да се стигне от всеки магазин до всеки друг. Разбира се, както хората, така и Ели, могат да се движат и в двете посоки по коридорите.

Изход

На стандартния изход изведете едно цяло число – броя хора в оптималното подмножество от коридори, което Ели може да избере.

Решение: Построяваме граф с върхове магазините и редица-гачените коридори със съответно тегло.

Използваме алг. на Крускал с модификация:
при добавяне на редица да запазваме теглото му в обща сума.

Заг Още една задача с леко модифициран алгоритъм на Крускал:

Задача В3. РАЗШИРЯВАНЕ НА КАНАЛИ**Автор: Георги Георгиев (Скелета)**

В страната Водландия има n езера (номерирани от 1 до n) и m канала между тях, като за всеки канал е известна ширината му в метри. Плаването по каналите може да се извършва и в двете посоки. Известно е, че лодка с ширина 1 метър може да достигне всяко езеро, тръгвайки от езеро с номер 1.

Напишете програма **channels**, която изчислява минималния брой канали, които трябва да бъдат разширени, така че лодка с ширина k метра да може да направи пътешествие между всеки две езера (лодката може да се придвижи от едно езеро до друго, ако нейната ширина е по-малка или равна на ширината на канала, свързващ езерата).

Вход

На първия ред са записани целите числа n и m ($1 < n \leq 1000$, $1 < m \leq 100000$).

Следват m реда, на всеки от които са записани по три цели числа i , j и w , показващи, че между езера i и j ($1 \leq i, j \leq n$) има канал с ширина w ($1 \leq w \leq 200$).

На последния ред на входа е записано цялото число k ($1 \leq k \leq 200$).

Изход

На един ред на стандартния изход програмата трябва да изведе едно цяло число, равно на най-малкия брой канали, които трябва да бъдат разширени.

Заг. (20)

НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА**Областен кръг, 20 март 2015 г.****Група В, 9-10 клас****ЗАДАЧА В2. ШАРЕНИ ТОПЧЕТА****Автор: Младен Манев**

Петърчо има N дървени топчета, номерирани с числата от 1 до N . Той трябва да боядиса всяко от тях в някакъв цвят. Напишете програма **balls**, която по зададени M двойки топчета, които трябва да бъдат едноцветни, намира най-големия брой цветове, които Петърчо може да използва за боядисването на всички топчета.

Вход

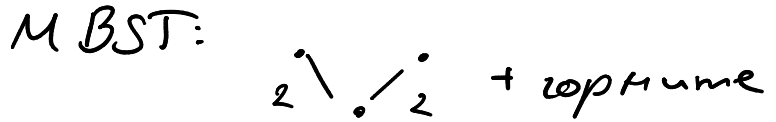
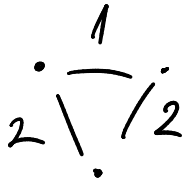
На първия ред на стандартния вход са зададени числата N и M . Следват M реда, на всеки от които са записани по две числа – номерата на две топчета, които трябва да бъдат едноцветни. Между дадените двойки топчета може да има повтарящи се.

Изход

На стандартния изход програмата трябва да се изведе най-големия брой цветове, които Петърчо може да използва за боядисване на топчетата.

- MBST - Minimum Bottleneck Spanning Tree
 $\rightarrow$ най-тежкото ребро да е възможно най-малко

Π_P



$$MST_s \subseteq MBST_s$$

Идея за линейен алг. за MBST:

1. $A \leftarrow$ множество от ребра с тежи наг медианата
2. $B \leftarrow$ останали ребра
3. Ако $G(V, B)$ е свързан
4. изв. рекурсивно за него
5. иначе
6. построи компонентен граф за $G(V, B)$
7. изв. се рекурсивно за тези компоненти
8. добавяй ребра от A докато не се свържат компонентите

$$T(n, m) = T\left(\frac{m}{2}\right) + \Theta(n)$$

$$\rightarrow T(m) = O(m \lg m)$$

// Carverini MBST

Най-квн нзмнча в графи

- SSShP

Алгоритми, които се използват:

- Dijkstra - използва приоритетна опашка от редра

- Работи само за неотрицателни тегла!

- Подобен на алг. на Prim

- $T(n, m) = n \cdot T_{EM} + m \cdot T_{DK} = \dots$

Extract Min Decrease Key

1) $= n \cdot O(\lg n) + m \cdot O(\lg n) = O((n+m) \lg n) \rightarrow \text{Binary Heap}$

2) $= n \cdot O(\lg n) + m \cdot O(1) = O(n \lg n + m) \rightarrow \text{Fibonacci Heap}$

// За една кутура. Fibonacci Heap не се разделя в курса.

- Bellman - Ford

- Работи за отрицателни тегла, но без отрицателни цикли.

↓
Ако има отрицателни цикли, не работи. (и връща False)
например

- $T(n, m) = O(n \cdot m)$

Зад За специални видове графи има и по-оптимални алг.

Например за DAG има линейно по време.

Заг Има n студенти на изпит. Знаем кои могат да си подсиават, като за всеки двойка знаем дали това е опасно или не. За дадени двама студенти да се намери най-безопасният път за подсиаване.

Решение: Постр. граф с върхове съответстващи на всеки студент.

Ребро m_{ij} два върха има ТСТК съответстващите студенти могат да си подсиават, с тегло 0, ако това е безопасно и 1 иначе.

Търсим най-кратък път от u до v , съответно за двамата студенти, с Дейкстра (например)

Заг Даден е граф с n сгради, k от които са болници, останалите - къщи. ($k \leq n$). Знаем кои сгради с кои други е свързани и колко е дълъг пътът. Да се намери кой къща е най-отдалечена от болница.

Решение: Искаме да намерим къщата, за която най-краткото разстояние до болница (когато е) е максимално.

Нека имаме граф с n върха - сградите и m ребра - директните връзки m_{ij} макс. $O(n^2m)$ доп. работи за построяване

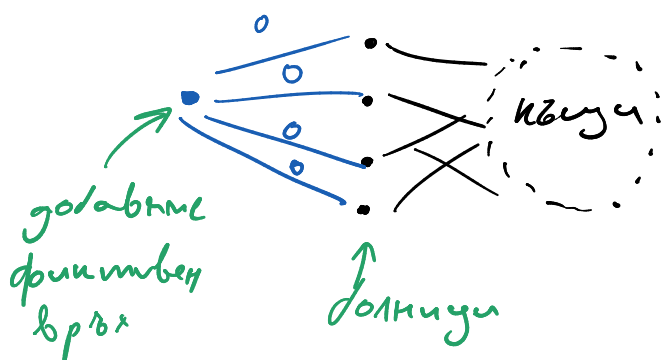
И. От всяка къща пускаме Dijkstra до всяка болница

$$T(n, m, k) = \underbrace{(n-k)}_{\text{брой къщи}} \cdot \underbrace{O(n \lg n + m)}_{\text{Dijkstra}} = O(n^2 \lg n - k n \lg n + mn - km) = O(n^2 \lg n + mn)$$

II. От вс. болница до всяка къща Dijkstra.

$$T(n, m, k) = \underbrace{k}_{\text{бр. болници}} \cdot \underbrace{O(n \lg n + m)}_{\text{Dijkstra}} + \underbrace{O(n-k)}_{\text{намеряване на max измежду къщите}} = O(k n \lg n + km + n - k)$$

III н. Можем да добавим един ориентиран връх:



За всеки връх най-къс път до ориентиран връх е най-къс път до долнища.

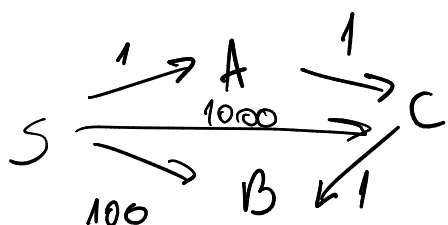
$$T(n, m, k) = O((n+1) \lg(n+1) + (m+k)) + O(n-k) =$$

$$= O(n \lg n + m + \dots)$$

Зад. 3 Подготвяйки лекцията си по ДАА за алгоритъма на Дийкстра, доцент Алгоритмов получава блестяща идея: да направи алгоритъма на Дийкстра хем по-бърз, хем по-лесен за имплементиране, като замени приоритетната опашка с обикновена опашка FIFO (First-In, First-Out). Какво бихте казали за тази идея?

(поправка 30.08.2022)

Решение:



Ако тръгнем път от S до B:

От опашката:

1. Изваждане S $\Rightarrow$ $d(A)=1$
 $d(B)=100$
 $d(C)=1000$

2. Изваждане B $\Rightarrow$ няма промяна

3. Изваждане C $\Rightarrow$ няма промяна

4. Изваждане A $\Rightarrow d(C)=2$, останалите - без промяна

Така получаваме: $d(A)=1$ ✓
 $d(B)=100$ ✗
 $d(C)=2$ ✓

// Тъй като използваме обикновената опашка, нищо не задължава алгоритъма да извади A преди B и C

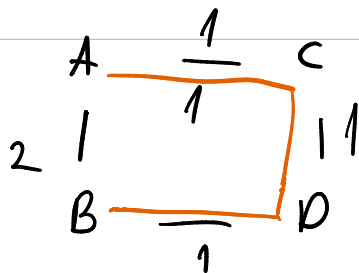
и да получи правилния резултат за разстоянието.

Узрм 27.06.2015:

15 т. Зад. 4 Докажете или опровергайте, че за всеки свързан тегловен неориентиран граф $G = (V, E)$ с тегловна функция w е вярно, че за всеки два върха $u, v \in V$ е изпълнено $\text{dist}_T(u, v) = \text{dist}_G(u, v)$, където

- T е графът, който връща Алгоритъмът на Крускал с вход (G, w) .
- $\text{dist}_T(u, v)$ е най-малката претеглена дължина на път между u и v в T
- $\text{dist}_G(u, v)$ е най-малката претеглена дължина на път между u и v в G
- претеглена дължина на път p в някакъв тегловен граф е сумата от теглата на ребрата на p .

Решение:



— МТД

$$\text{dist}_T(A, B) = 3$$

$$\text{dist}_G(A, B) = 2$$

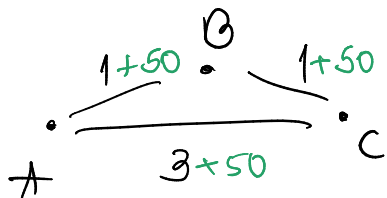
Заг Ако в тегловен граф е неориент. тегла всички теми се увеличават с положителна константа.

- Ще се запази ли МТД? (камо множество от подгра)
- Ще се запазят ли най-късите пътища?
(камо последователност от върхове)

Решение:

— МТД все се запази, защото подредбата на теглата не се променя.

— Най-късите пътища не се запазват, например:



$$\Rightarrow 51 + 51 > 53$$

и пътя м/у A-C е гръз