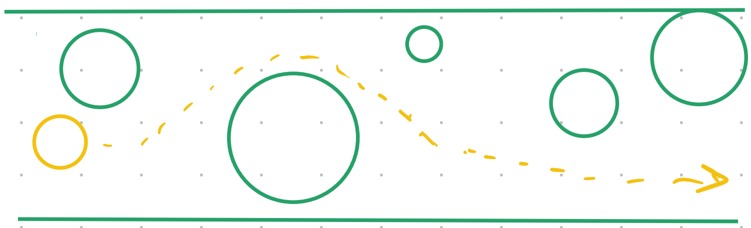
ДАА семинар 11

Зад. В двумерното пространство е дадена тръба с препятствия – кръгове с център (x,y) и радиус r. Знаем и ширината на тръбата. Хамстер е в кръг с даден радиус. Може ли хамстерът да премине от единия край до другия край на тръбата?



Решение: Построяваме граф с по 1 връх за всяко препятствие, както и 2 върха за стените.

Между два върха има ребро ТСТК разстоянието между съответните геометрични обекти е по-малко от размера на хамстеровия кръг. (диаметъра)

Оттук, хамстерът може да премине ТСТК няма път между върховете съответно за двете стени.

$$T(n) = \underbrace{\Theta(n^2)}_{\text{граф}} + \underbrace{\Theta(n^2)}_{BFS} = \Theta(n^2)$$

$$S(n) = \underbrace{O(n^2)}_{\text{граф}} + \underbrace{\Theta(n^2)}_{BFS} = O(n^2)$$

DAG и Топосортиране

- DAG = Directed Acyclic Graph
  - поне един източник, поне един сифон

- Топологическо сортиране за $G = (V, E)$

  биекция $h: V \to \{1..n\}$, т.че $\forall (u,v) \in E \;\; h(u) < h(v)$

- G е DAG $\iff \exists$ топосортиране

- Можем да намерим топологическо сортиране за $\Theta(n+m)$ време и $\Theta(n)$ памет

<u>Пример</u>. Намиране на най-дълъг път в DAG.

```
LongestPathDAG( G=(V,E) - DAG)
  A ← Toposort(G)
  d[1..n]   - ако е попълнена d[i] съдържа дължина
              на най-дълъг път, започващ в съотв. връх.
  foreach v ∈ reverse(A)
     d[v] ← 0
        foreach u ∈ adj[v]
             d[v] ← max(d[v], d[u]+1)
  return max(d[1..n])
```

//Лесно може да се модифицира да връща върха

- Как модифицираме алг., ако G е тегловен?

Зад. Колко най-много ребра може да има в ДАГ?

Решение: $(n-1) + (n-2) + \ldots + 1 + 0 = \frac{n(n-1)}{2} = \Theta(n^2)$

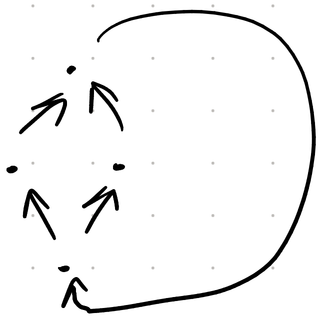
Нека $G=(V,E)$ е ориентиран граф. Компонентният му граф е ДАГ (ако има цикли, те са в една силносвързана компонента)

Зад. Докажете или опровергайте:

а) 1 SSC $\Leftrightarrow \exists$ HC

SSC – силносв. комп.
HC – Хамилтонов цикъл
HP – Хам. път

($\Rightarrow$) Не е вярно:



→ За SSC може да има припокриващи се пътища.
За HC не може.

($\Leftarrow$) Да, вярно е:

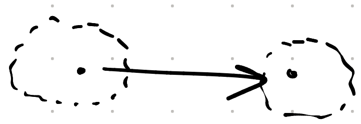
От всеки връх може да стигнем до всеки друг, използвайки цикъла

Т.е.: $\exists$ HC $\Rightarrow$ 1 SSC

б) 1 SSC $\Leftrightarrow \exists$ HP

($\Rightarrow$) Не, като в а)

($\Leftarrow$) Не, контрапример:



Има HP, но 2 SSC

$A \Rightarrow B \iff \neg B \Rightarrow \neg A$

в) >1 SSC $\iff \nexists$ HC

($\Rightarrow$) вярно – от а) ? Контрапозиция? Да.

($\Leftarrow$) Не – същия пример като в а)

ДР Зад. В ОАГ има единств. сорт. $\iff \exists$ HP

Зад. Даден е речник над думи от неизвестна азбука, за който се твърди, че е в лексикографска наредба.

Да се провери дали това е така.

Пр1.	Пр2	Пр3
abc	abc	7*&++
acc	acc	a4%%
baby	bdz	a4&%
bdz	bad	&wxx
cafe	ccc	&~af
	down	~oops

Пр1:

b преди c
a преди b
a преди d

→ напр: yadbcz

⇒ Да

Пр2:

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

$$\begin{tikzcd} a \arrow[r] & b \arrow[r] & c \arrow[dl] \\ & d \arrow[ul] & \end{tikzcd}$$

⇒ Не, има цикъл

Пр3:

?

Решение: Искаме да проверим дали има азбука, която да се получава от наредба 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 тстк графът е ацикличен.

Графът е ориентиран. Ако няма цикли е DAG, т.е. може да продължим с топологическо сортиране.

Ако има цикъл, ще го открие; ако има азбука, ще ни я даде.

// Линейна наредба → топологическо сортиране

- MST - Minimum Spanning Tree

  - Kruskal - започва от гора от върхове, които обединява чрез Union-Find, обработвайки ребрата по тегло.

    - $T(n,m) = \theta(m \lg n)$

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

    - Find - дава ни лидер на компонентата // Чрез 2 масива;

      Union by rank

      По-подробно в лекционните записки

  - Prim - започва от произволен връх и алчно добавя ребра

Зад

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 $\mathbf{A_i}$ $\mathbf{B_i}$ $\mathbf{P_i}$, указващи, че между магазините с индекси $\mathbf{A_i}$ и $\mathbf{B_i}$ има коридор, по който минават $\mathbf{P_i}$ 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 ребра - дадените коридори със съответно тегло.

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

Зад Още една задача с леко модифициран алгоритъм на Крускал:

НАЦИОНАЛЕН ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА
Шумен, 23 – 25 ноември 2012 г.
Група B, 9 – 10 клас

Задача B3. 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.

Зад. DP

НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Областен кръг, 20 март 2015 г.
Група B, 9-10 клас

ЗАДАЧА B2. 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.

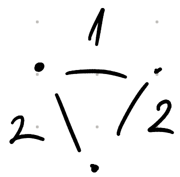
Изход

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

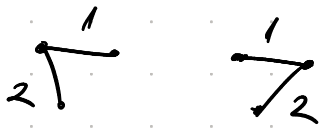
- MBST - Minimum Bottleneck Spanning Tree

→ най-тежкото ребро да е възможно най-леко

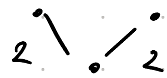
Пр



MST:



MBST:



+ горните

MSTs $\subseteq$ MBSTs

Идея за линеен алг. за MBST:

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

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

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

// Camerini MBSA