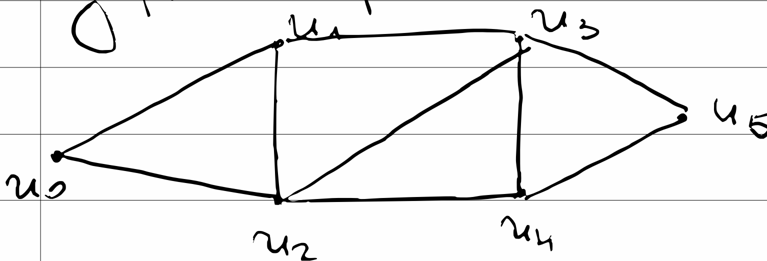
(302) Да се намери покриващо дърво на графа:



a) с bfs;

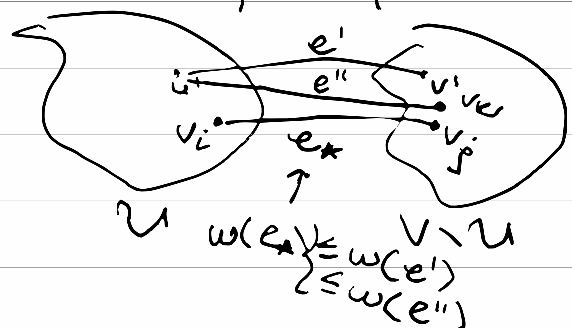
b) с dfs.

Optimal spanning tree

```
Kruskal(G, w):
    E_T = { }
    Sort edges E by inc. weight using w.
    for all edges {u,v} ∈ E:
        if adding {u,v} to E_T does not
                form a cycle:
            add edge to E_T
    return E_T
```

Най-проста имплементация.
Оповава се на th за съгласуваното мн-во.

$\boxed{\text{Th}}$ Нека $G = \langle V, \mathcal{E} \rangle$ е НГ, свързан с ценова ф-ия на ребрата $w: \mathcal{E} \to \mathbb{R}$ и $\emptyset \neq U \subsetneq V$. Нека $e = \{v_i, v_j\} \in \mathcal{E}$ е т.е. $v_i \in U$, $v_j \notin U$ и има мин. тегло измежду всички такива ребра. Тогава съществува MST $T_0 = \langle W, \mathcal{E}_0 \rangle$, т.е. $e \in \mathcal{E}_0$.



```
Prim-Jarnik(G, w, s)
  forall v ∈ V
    cost(v) = ∞
    parent(v) = null
  cost(s) = 0
  PQ = makePQ(V0, cost)  // строи приор. опашка спрямо cost
  while PQ is not empty:
    u = popMin(PQ)  // връща върха с най-малък ключ
```

намерили сме връх, за който можем да подобрим цената му

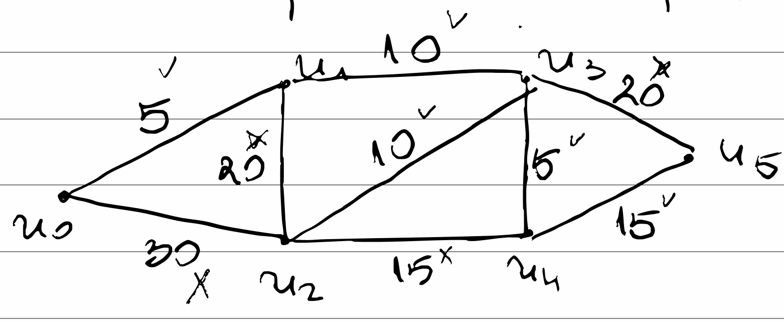
for each $\{u, v\} \in E$:

    if cost(v) > w(u, v):
        cost(v) = w(u, v)
        parent(v) = u
        decreaseKey(PQ, v)

Използва сигнал към опашката, че cost ст-та на v е било намалено и предизвиква пренареждане на PQ спрямо обновения масив cost.

relax(u, v, w, cost, parent, PQ)

392) Да се намери MST на графа.



a) с Kruskal:

$u_0u_1 - 5$ ✓

$u_3u_5 - 5$ ✓

$u_1u_3 - 10$ ✓

$u_2u_3 - 10$ ✓

$\cancel{u_2u_4 - 15}$ → става цикъл

$u_4u_5 - 15$ ✓

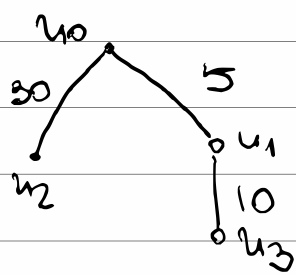
$u_3u_5 - 20$

$u_1u_2 - 20$

$u_0u_2 - 30$

min тегло = 45

b) Прим-Ярник:



SPST

Дървета на най-късите пътища

* Ако няма тегла на ребрата, казахме, че bfs строи от даден връх до всички други.
* Ако има положителни тегла и
  - всички тегла са равни -> ни bfs;
  - има поне две различни -> Dijkstra.
* Ако графът е DAG (directed acyclyc graph), има по-хубави алгоритми.
* Ако сред теглата има отрицателни, има други алгоритми, които строят SPT.
* Dijkstra строи дърво на най-късите пътища от даден връх до всички други (работи за ДОГ и КНР).

! MST ≠ SPST !

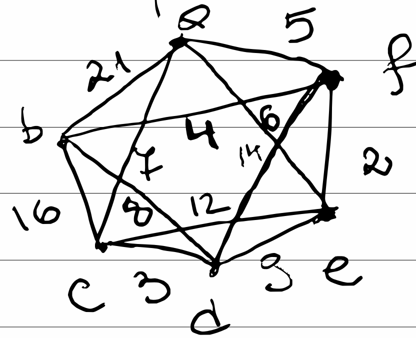
```
Dijkstra(G, w, s)
  for all v ∈ V:
      dist(v) = ∞
      parent(v) = null
  dist(s) = 0   // тривиален път от s до s
  PQ = makePQ(V, dist)
  while PQ is not empty:
      u = deleteMin(PQ)
      for all edges {u,v} ∈ E:
          if dist(v) > dist(u) + w(u,v):
              dist(v) = dist(u) + w(u,v)
              parent(v) = u
              decreasekey(PQ, v)
```

взимаме върха, до който стигаме най-бързо

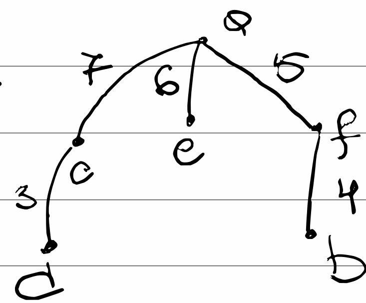
намерили сме по-лек път до v през u

Тв | Всеки начален отрез на най-лек път от s строен от Dijkstra е също най-лек път.

Зад) Постройте MST на



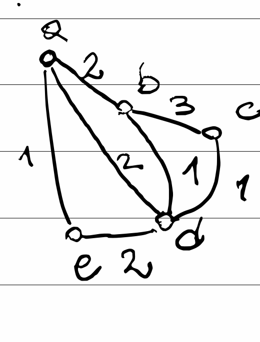
с начало a.



Зад) КНГ $G = \langle V, R \rangle$ с м-во върхове $V = \{a, b, c, d, e\}$ е зададен с матрица M:

M =

∞	2	∞	2	1
2	∞	3	1	∞
∞	3	∞	1	∞
2	1	1	∞	2
1	∞	∞	2	∞



с елементи $m_{ij} = \begin{cases} \text{"}\in N\text{"}, & \text{ако } \langle i,j \rangle \in R \\ \infty, & \text{ако } \langle i,j \rangle \notin R \end{cases}$

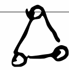
а) По Kruskal MST без корен;

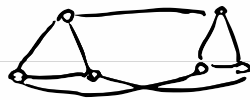
b) По Prim-Jarnick MST с корен $r=3$.

c) SPST с Dijkstra с корен $r=1$

d) SPST с корен $r=3$, ако теглата на всички ребра е 1.

(302) $(G_n)_{n=0}^{\infty}$ редица от графи.

$G_0 = K_3$ 

$G_1 \longrightarrow$ 

G_{n+1} = две копия на G_n като м/у копията има ребро, които свързват съответните върхове на копията.

За зададен $G_n = \langle V_n, E_n \rangle$, колко са $|V_n|$ и $|E_n|$

Сега $|V_0| = 3$.

$|V_{n+1}| = 2.|V_n|$.

Нека $a_n = |V_n|$.

$$\begin{cases} a_0 = 3 \\ a_{n+1} = 2.a_n \end{cases}$$

Тогава $a_n = 3.2^n$.

За $|E_0| = 3$, $|E_n| = 2.|E_{n-1}| + |V_{n-1}|$.

Нека $b_n = |E_n|$.

$$\begin{cases} b_0 = 3 \\ b_n = 2.b_{n-1} + 3.2^{n-1} \end{cases}$$

$\{2\}_M \cup \{2\}_M = \{2,2\}_M$

$b_n = (A_1 + n.A_2).2^n$

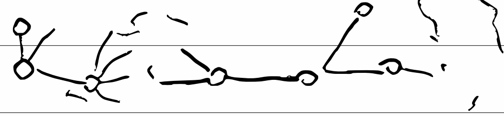
$b_0 = 3 = A_1$

$b_1 = 9 = 2.3 + 2.A_2 \iff A_2 = 3/2$

Значи $b_n = 3.n.2^{n-1} + 3.2^n$.

(3зад) Даден е граф $G = \langle V, E \rangle$.
Реброто е се нарича мост/свързващо/критично за G, ако при отстраняването му се увеличава броят на свързаните компоненти в G. Докажете, че ако всички върхове на графа са от четна степен, то той не съдържа мост.

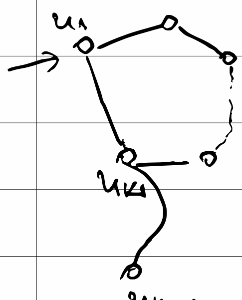
Нека за $(\forall v \in V)[d(v)$ е четно$]$.
Нека $C_1, \ldots, C_k$ са свързаните компоненти на G. Тогава във всяка една от свързаните комп. има Ойлерова верига.
Нека C_i разгл. за $i \in \{1, \ldots, k\}$.
Ако допуснем, че в нея има мост, то тръгвайки от произволен връх,



за да обиколим всички ребра трябва да минем и по него, но няма да можем да се върнем обратно към началния връх на веригата ↯.

(302) Нека $G=\langle V,E\rangle$ е свързан НГ с $|V|=n$. Нека $u_1,\dots,u_k$ е най-дълъг прост път в G за $k<n$.
Докажете, че u_1 и u_k не са съседни.

Нека допуснем, че са съседни.
Нека използваме, че G е свързан и Б.О.О. има връх u_{k+1} излизащ от u_k.



Но тогава, ако пуснем алгоритъм строящ покриващо дърво от u_{k+1}, то ще получим пък: $u_{k+1}u_k u_1 \dots u_{k-1}$ прост път, който е по-дълъг от най-дългия прост път $u_1,\dots u_k$. Абсурд!

(308) $G = \langle V, E \rangle$ е КНГ, в който $(\forall v \in V)[d(v) \geq d]$, където $d \in \mathbb{N}_{>1}$.

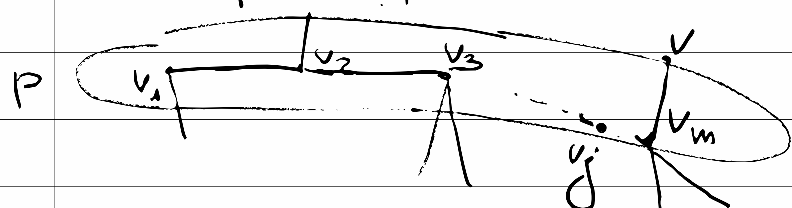
a) Докажете, че в G има цикъл;

b) Докажете, че в G има цикъл с дължина поне $d+1$.

---

a) Нека $|V| = n$.

Нека p е произволен макс. път в G.



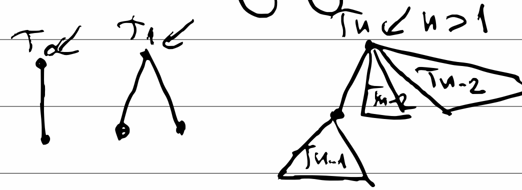
Т.к. $d(v_m) \geq 2$, то значи той е свързан с ребро с поне още един връх.

Сл.1 Ако е извън p този връх, ще получим по-дълъг път pv, но p е с макс. дължина ↯.

Сл.2 Значи v_m е свързан с ребро за някой връх от пътя p. Т.к. вече ще има път м/у тях, успяхме да заформим цикъл.

b) Нека $p = v_0 \dots v_k$ е най-дълъг път. Т.к. $d(v_0) \geq 2$, то той ще има още съседи, които са вътре в p (иначе $\nleq$ с избора на p). Тогава излиза, че всички съседи на v_0 са в пътя p.
Нека i е максимално индекс на връх съсед на v_0 от пътя p.
Т.к. $d(v_0) \geq 2$, то $i \geq deg(v_0)$, иначе няма да се поберат съседите на v_0. Тогава има цикъл $v_0 \dots v_i v_0$, който е с дължина поне $d(v_0)+1$ на брой ребра, но $d(v_0) = d$ т.е. намерихме цикъл с дължина поне $d+1$.

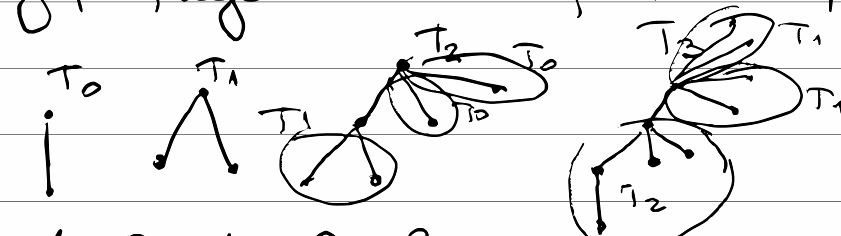
302) $(T_n)_{n=0}^{\infty}$ – редица от коренови дървета



Намерете броя на вътрешните върхове за $n \in \mathbb{N}$: T_n.

Нека a_n = # вътрешни върхове в дървото T_n.

def| Листо в дърво е връх от степен 1. Ще се уговорим, че в кореновото дърво, ще считаме корена за вътрешен.



$a_0 = 1$ $a_1 = 1$ $a_2 = 2$

$a_3 = 2 + 2.1 - 1 = 3$

Значи $a_n = a_{n-1} + 2a_{n-2} \underline{-1}$ заради единия от корените

$\{-1, 1, 2\}_m$

$a_n = A_1.(-1)^n + A_2.1^n + A_3.2^n$ общ вид

$$\begin{cases} a_0 = 1 = A_1 + A_2 + A_3 \\ a_1 = 1 = -A_1 + A_2 + 2A_3 \\ a_2 = 2 = A_1 + A_2 + 4A_3 \end{cases}$$

$$A_1 = 1/6$$

$$2 = 2A_2 + 3A_3$$

$$3 = 2A_2 + 6A_3$$

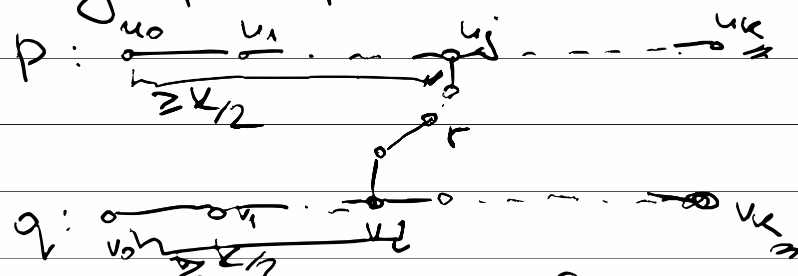
$$A_2 = 1/2$$

$$-1 = -3A_3 \Leftrightarrow A_3 = 1/3$$

$$\text{ОТГ: } a_n = \frac{(-1)^n + 3 + 2^{n+1}}{6}$$

(302) Нека G е дърво. Покажете, че има един връх общ за всички най-дълги пътища в G

Ако имаме само един най-дълъг път е тривиално.
Ако са два, то вече в отделна задача показахме, че ще имат общ връх. Идеята беше за доп., че не е така. Нека вземем два най-дълги прости пътища нямащи общ връх p и q

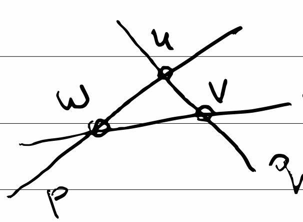


Т.к. G е дърво, значи G е свързан при това ще е единствен пътя свързващ всеки два върха ~> ще има връх от p : u_i и връх от q : v_i, т.е. ги свързва път r.

В Г не се срещат други върхове от p и q, защото иначе ще имаме цикъл. БОО нека $d \geq \lceil \frac{k}{2} \rceil$.

Тогава пътя $u_0 \dots u_j \Gamma v_i \dots v_0$ е с дължина поне $\lceil \frac{k}{2} \rceil + \lceil \frac{k}{2} \rceil + 1 \geq \frac{k}{2} + \frac{k}{2} + 1 = k+1$ Абсурд с избора на p и q като най-далечни.

Сега, ако имаме поне 3 прости най-дълги пътища, нека док., че имат общ връх. Знаем, че всеки два от тях имат общ връх и нека ги вземем три.

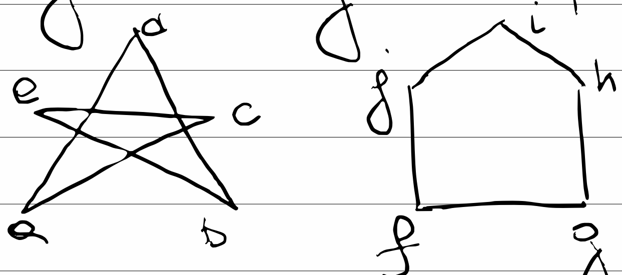


Тогава има пресечни точки w, u, v и те ще са част от цикъл, но G е дърво $\Rightarrow$ Абсурд!

def | Нека $G_1 = \langle V_1, E_1 \rangle$ и $G_2 = \langle V_2, E_2 \rangle$ са два КНГ. $G_1 \cong G_2$, ако съществува функция $f: V_1 \to V_2$, такава че:

- f е биекция на V_1 в/у V_2;
- за всеки два върха $a, b \in V_1$, то $\{a, b\} \in E_1 \Longleftrightarrow \{f(a), f(b)\} \in E_2$

? Едни и същи ли са графите?



Пр, нека $f: \{a, b, c, d, e\} \to \{i, j, h, f, g\}$
т.ч. $f(c) = j$, $f(e) = h$, $f(d) = i$,
$f(a) = f$, $f(b) = g$.
f е биекция тривиално и запазва релациите „ребро" м/у върховете.
Значи те са изоморфни.

Не може винаги толкова конкретно да го правим това доказателство за графи с много върхове.

Друго характеризиращо ≅ м/у графи са и следните условия:

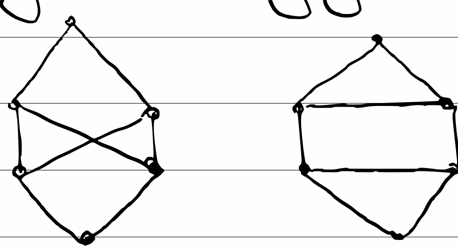
(i) Да имат равен брой върхове;

(ii) Да имат равен брой ребра;

(iii) Ако се наредят в две редици по големина степените на върховете на двата графа, то двете редици трябва да съвпадат.

(iv) Да имат един и същ брой на цикли от дадена дължина.

прим:



Изоморфни ли са?

Не - различават се в това, че един от тях съдържа цикъл C_5.

(30зад) Пътна мрежа е м-во градове и шосета, като всяко шосе свързва два града. Пътната мрежа на някаква държава е такава, че от всеки град може да се стигне до всеки, но по единствен начин, и освен това броят на шосетата е четно число. Докажете, че има поне един град, от който излизат четен брой шосета.

Имаме дърво с четен брой ребра.
Искаме да видим, че $(\exists v \in V)[d(v)$ е четно$]$

Нека $|V|=n$. Тогава $|E|=n-1$, т.е. n е нечетно число. Но за всеки граф броят на върховете от нечетна степен е четно число $\rightsquigarrow$ ↯.