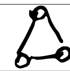
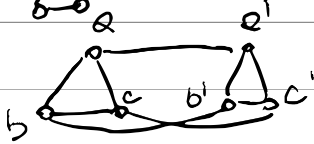
300. $(G_n)_{n=0}^{\infty}$ редица от графи.

$G_0 = K_3$ 

$G_1 \longrightarrow$ 

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

Задача: $G_n = \langle V_n, E_n \rangle$, колко са $|V_n|$ и $|E_n|$

G_0: $|V_0| = 3$ $|E_0| = 3$

G_1: $|V_1| = 6$, $|E_1| = 6 + 3 = 9$

$a_n \leftrightharpoons |V_n|$.

$$\left| \begin{array}{l} a_0 = 3 \\ a_{n+1} = 2 \cdot a_n \end{array} \right\} \rightsquigarrow \boxed{a_n = 3 \cdot 2^n}$$

$$b_n \leftrightharpoons |E_n| \rightsquigarrow \left| \begin{array}{l} b_0 = 3 \\ b_{n+1} = 2 \cdot b_n + \overbrace{a_n}^{3 \cdot 2^n} \end{array} \right.$$

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

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

$$b_n = (A_1 + n \cdot A_2) \cdot 2^n$$

$$\begin{cases} b_0 = 3 = A_1 \cdot 2^0 = A_1 \leadsto A_1 = 3 \\ b_1 = 3 = \underbrace{A_1 \cdot 2}_{6} + A_2 \cdot 2 \end{cases}$$

$$A_2 = 3/2$$

$$b_n = \left(3 + n \cdot \frac{3}{2}\right) \cdot 2^n = 3 \cdot 2^n + n \cdot 3 \cdot 2^{n-1}$$

Optimal spanning tree

Kruskal(G, w):

  $E_T = \{\}$

  Sort edges E by inc. weight using w.

  for all edges $\{u, v\} \in E$:

    if adding $\{u, v\}$ to E_T does not form a cycle:

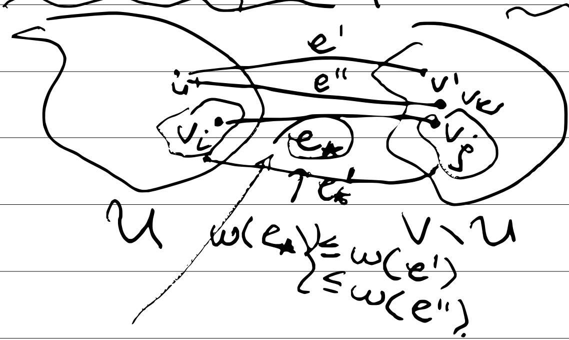
      add edge to E_T

  return E_T

Най-проста имплементация.

Опозава се на th за съгласуваното мн-во:

$\boxed{\text{Th}}$ Нека $G = \langle V, E \rangle$ е НГ, свързан с ценова ф-я на ребрата $\omega : E \to \mathbb{R}$ и $\emptyset \neq U \subsetneq V$. Нека $e = \{v_i, v_j\} \in E$ е т.е. $v_i \in U$, $v_j \notin U$ и има мин. тегло измежду всички такива ребра. Тогава съществува MST $T_0 = \langle W, E_0 \rangle$, т.е. $e \in E_0$.



$e_0 < e_1 < e_2$ - ——

Prim-Jarnik(G, w, s)

for all $v \in V$

  cost(v) = ∞

  parent(v) = null

cost(s) = 0

PQ = make PQ(V_0, cost) // строи приор. опашка спрямо cost

while PQ is not empty:

  u = popMin(PQ) // връща върха с най-малък ключ

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

намерили сме връх, за който можем да подобрим цената му

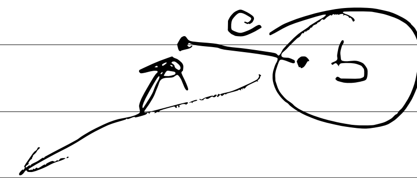
    if cost(v) > w(u, v):

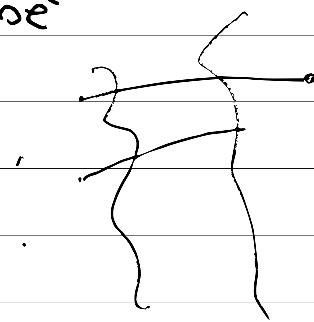
      cost(v) = w(u, v)

      parent(v) = u

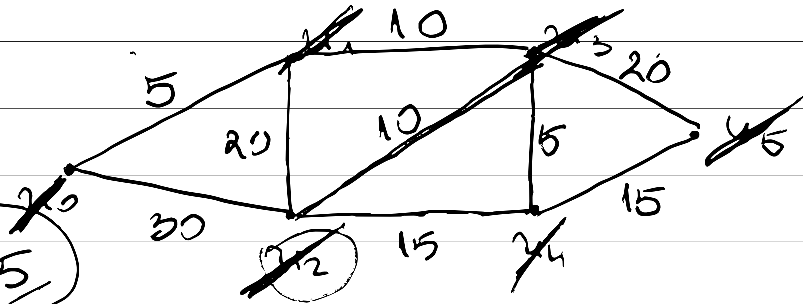
      decreaseKey(PQ, v)

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





392) Да се намери MST на графа.



$u_3u_4 - 5$

$u_0u_1 - 5$

$u_1u_3 - 10$

$u_2u_3 - 10$

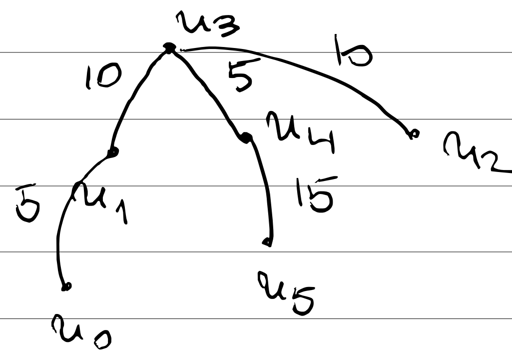
$u_2u_4 - 15$

$u_4u_5 - 15$

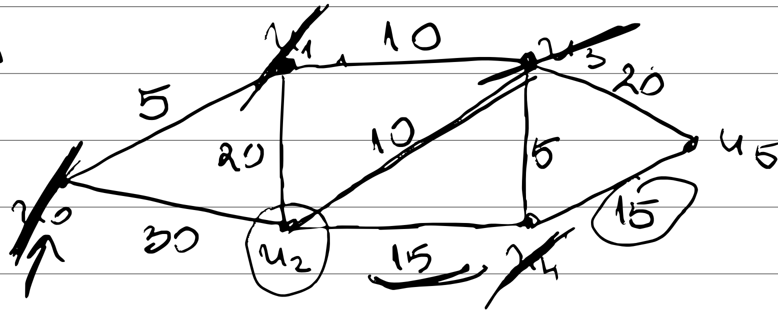
$u_3u_5 - 20$

$u_1u_2 - 20$

$u_0u_2 - 30$



Prim-Jarnik



стъпка 0: u_0: $cost(u_1) = 5$

$cost(u_2) = 30$

стъпка 1: u_1: $cost(u_2) = 20$

$cost(u_3) = 10$

стъпка 2: u_3: $cost(u_2) = 10$

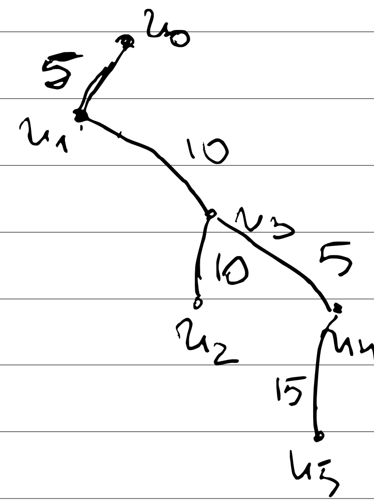
$cost(u_4) = 5$

$cost(u_5) = 20$

стъпка 3: u_4: $cost(u_5) = 15$

стъпка 4: u_2

стъпка 5: u_5



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 -> най bfs;
  - има поне две различни -> Dijkstra.
* Ако графът е DAG (directed acyclyc graph), има по-хубави алгоритми.

* Ако средтеглото има отрицателни, има други алгоритми, които строят SPT.
* Dijkstra строи дърво на най-късите пътища от даден връх до всички други (работи за ПОГ и КНГ).

! MST ≠ SPST !

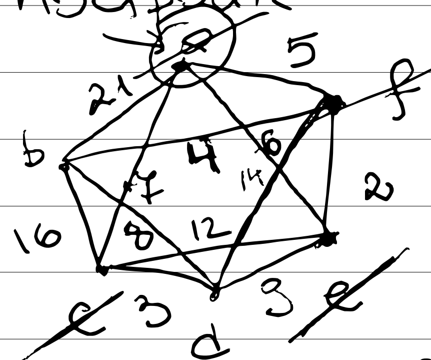
Тв| Всеки начален отрез на най-къс път от s строен от Dijkstra е също най-къс път.

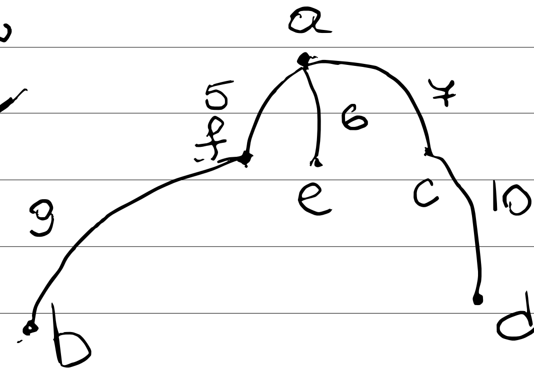
```
Dijkstra(G, w, s)
  for all v ∈ V:
      dist(v) = ∞
      parent(v) = null
  dist(s) = 0          // тривиален път от s до s
  PQ = makePQ(V, dist)
  while PQ is not empty:
      u = deleteMin(PQ)
      for all edges {u,v} ∈ E:
          if dist(v) > dist(u) + w(u,v)
              dist(v) = dist(u)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

30 Постройте MST на графе с корнем a.





ст. 0: a

ст. 1: f

ст. 2: e

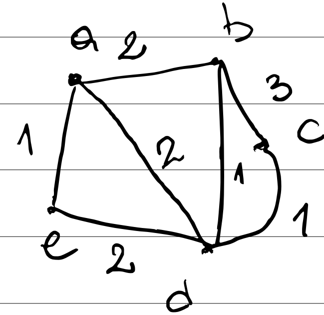
ст. 3: c

ст. 4: b

ст. 5: d

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

$$M = \begin{array}{|c|c|c|c|c|} \infty & 2 & \infty & 2 & 1 \\ \hline 2 & \infty & 3 & 1 & \infty \\ \hline \infty & 3 & \infty & 1 & \infty \\ \hline 2 & 1 & 1 & \infty & 2 \\ \hline 1 & \infty & \infty & 2 & \infty \end{array}$$



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

a) По Kruskal MST без корен;

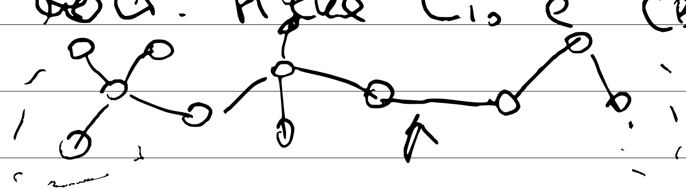
b) По Prim-Jarnick MST с корен $r = 3$.

c) SPST с Dijkstra с корен $r = 1$

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

(3.9.3) Даден е граф $G = \langle V, E \rangle$.
Реброто е се нарича мост/срязващо/ 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 $(\forall v \in V)$ [„$d(v)$ е четно"].
Нека $C_1, \dots, C_k$ са комп. на свързаност на G.
Тогава за всяко $i \in \{1, \dots, k\}$ всички върхове в C_i отново са от четна степен.
Нека допуснем, че за някое $i \in \{1, \dots, k\}$ в C_i има мост. Нека C_{i_0} е свидетел.



По Th на Ойлер за C_{i_0} $\rightsquigarrow$ затворена Ойлерова верига $\lightning$

(С този мост, няма да можем да затворим веригата.

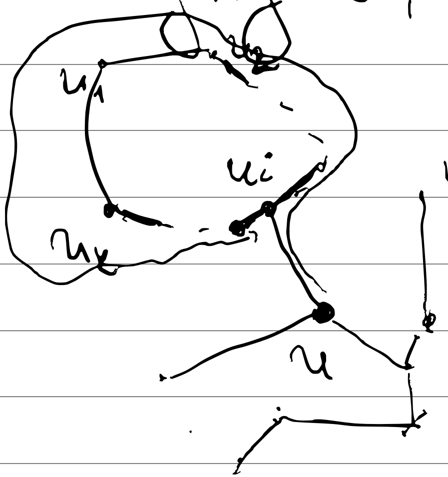
Махаме реброто $\rightsquigarrow$ 2 нови комп. на св. C_{i_0}' и C_{i_0}'', в които получаваме да имат точно по един връх от нечетна степен. $\rightsquigarrow$ $\lightning$ с $\sum_{v \in C_{i_0}'} d(v)$ е четно

(392) 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.

$p = u_1 u_2 \dots u_k$, $u_i E u_{i+1}$, $i \in \{1, \dots, k-1\}$

$k<n$.

Нека допуснем, че u_1 и u_k са съседни



Имаме път $u u_i \dots u_2 u_1 \dots u_k \dots u_{i+1}$ с $k+1 \le n$ върха. Абсурд, $u_1 \dots u_k$ е най-дълъг път.

Тогава u_1 и u_k излиза, че не трябва да са съседни.

Зад. $G = \langle V, E \rangle$ 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 $|V| = n$, $\underbrace{d \in \mathbb{N}_{>1}}_{d \geq 2,\ d \in \mathbb{N}}$, т.ч. $(\forall v \in V)[d(v) \geq d]$.

Нека p е макс. път в G

$d(v_0) \geq 2 \rightsquigarrow$ още едно ребро към връх: y

1) Върхът y не е от p.
Търсим макс. път: yp с $k+2$ върха !

2) Върхът y е от върховете в p.

Нека v_i е този връх, тогава $v_0 v_1 \ldots v_i v_0$ е цикъл.

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

Нека $p = v_0 \ldots v_k$ е най-дълъг път.

а) Всички ребра излизащи от върхове в p остават в p.

Нека за v_0 да разсъждаваме. $d(v_0) \geq d \geq 2$.

Нека v_j е т.ч. $\{v_0, v_j\} \in E$ и j е макс. инд.

От това, че а и $d(v_0) \geq d \rightarrow j \geq d$

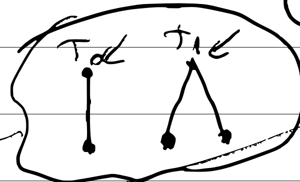
$v_0 v_1 \ldots v_j v_0 \rightarrow$ е с дължина поне $d+1$.

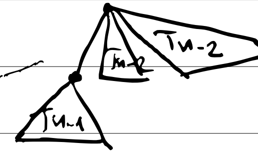
302 Докажете, че за всеки граф G (неориентиран без примки и краен), то

$$\underbrace{\chi(G)}_{\text{върховото хроматично число}} \le \frac{1}{2} + \sqrt{2m + \frac{1}{4}}, \quad |E| = m$$

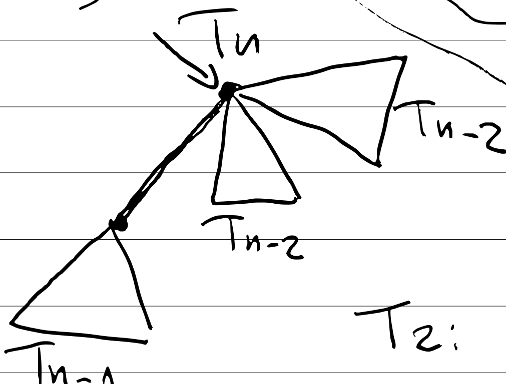
382 $(T_n)_{n=0}^{\infty}$ – редица от коренови дървета

T_n, $n \geq 1$





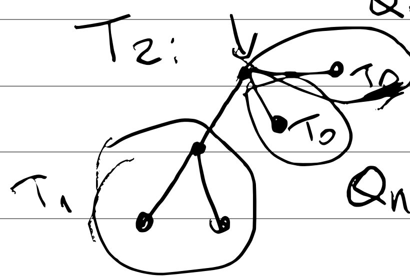
Намерете броя на вътрешните върхове за $n \in \mathbb{N}$: $T_n \to Q_n$



Ако коренът е вътрешен връх

$Q_0 = 1$

$Q_1 = 1$



$Q_2 = 2 = Q_1 + 2.Q_0 - 1$

$Q_{n+2} = Q_{n+1} + 2.Q_n - 1$

$$\begin{cases} a_0 = 1,\ a_1 = 1 \\ a_{n+2} = a_{n+1} + 2a_n - 1 \cdot 1^n \end{cases}$$

$$x^2 - x - 2 = 0$$

$$\{2, -1\}_M \cup \{1\}_M = \{2, -1, 1\}_M.$$

$$a_n = A_1 \cdot 2^n + A_2 \cdot (-1)^n + \underbrace{A_3 \cdot 1^n}_{A_3}$$

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

$A_1 = 1/3$

$A_2 = 1/6$, $A_3 = 1/2$

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

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

За пътя

(A) Нека допуснем, че има поне два най-дълги пътя в G т.ч. не се пресичат.

$p : v_0 — v_1 — \dots — v_j — \dots — v_k$

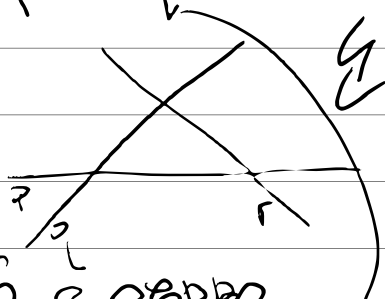
$q : u_0 — u_1 — \dots — u_i — \dots — u_k$

G е свързан $\leadsto$ има път r от u_i до v_j $\leadsto$ върховете в r не са от p и q освен v_i и v_j (иначе цикъл).

Нека $j, i \geq \lceil \frac{k}{2} \rceil$. Тогава $u_0 u_1 — u_i r v_j \dots v_1 v_0$ е път с $\geq \lceil \frac{k}{2} \rceil + \lceil \frac{k}{2} \rceil + 1 \geq k+1$ ↯ с максималност на p и q

Един най-добър път → V

Нека иначе има 3 най-добри пътя.
Нека вземем два най-добри пътя
р и q. За всеки трети най-добър път r
то той трябва да се пресече
едновременно с р и q.



б е добро

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

$(\exists v \in V)\ [d(v)$ е четно$]$

Върхове – градове, шосета – ребра, $|E| = 2.k$ за някое $k \in \mathbb{N}$.

$G = \langle V, E \rangle$, $|E| = 2k$, $k \in \mathbb{N}$

дърво

$|V| = n$, $|E| = n - 1 = 2.k$

$\Rightarrow$ n е нечетно

$$2\cdot|E| = \sum_{v\in V} d(v) = \sum_{v\in \underbrace{V_1}_{\text{нечетни}}} \underbrace{d(v)} + \sum_{v\in \underbrace{V_2}_{\text{четни}}} d(v)$$

Допускаме $V_2 = \emptyset$. Тогава $|V| = n$ ~ нечетно

$V = V_1 \rightsquigarrow |V_1| = n$

нечетен брой върхове от нечетна степен.

↯