(320) Нека $G = \langle V, E \rangle$ е свързан НГ и $w \in V$ е негов срязващ/критичен връх (след неговото отстраняване на w и ребрата, инцидентни с него, броят на свързаните компоненти на G се увеличава).

Нека G_1 е графът, получен от G чрез отстраняване на w и ребрата, инцидентни с него. Докажете, че графът $\overline{G_1}$ е свързан.

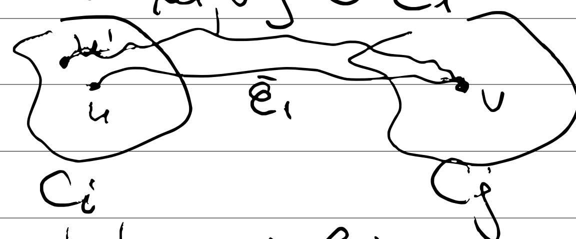
$$\overline{G}_1 = \langle \underbrace{V \setminus \{w\}}_{V_1}, \underbrace{V_1 \times V_1 \setminus E_1}_{\overline{E}_1} \rangle$$

НП
Свързан граф – м/у всеки два върха има път.

$\overline{G_1}$

Нека $C_1, \dots, C_k$ са свързаните компоненти на G_1.
различни

- Нека $i, j \in \{1, \dots, k\}$, $i \neq j \Rightarrow C_i, C_j$.
  Нека $u \in C_i$, $v \in C_j$. Тогава $\{u, v\} \notin E_1 \Rightarrow \{u, v\} \in \overline{E_1}$ (*)



- Нека $i \in \{1, \dots, k\} \Rightarrow C_i$.
  Нека $u, u' \in C_i \Rightarrow$ м/у u и u' има път в G_1.
  От (*) $\Rightarrow \{u', v\} \in \overline{E_1}$, $\{u', v\}, \{v, u\} \in \overline{E_1}$.
  Значи м/у u' и u има път и в $\overline{G_1}$.
  $\overline{G_1}$ е свързан.

def | Клика

Пълен граф т.е. $G = \langle V, E \rangle$

и $|E| = \binom{n}{2} = \frac{n.(n-1)}{2}$

За произволен граф имаме, че

$0 \leq |E| \leq \binom{n}{2}$.

Антиклика е граф без ребра

т.е. няма два върха, които да

са свързани с ребро $\rightarrow |E| = 0$.

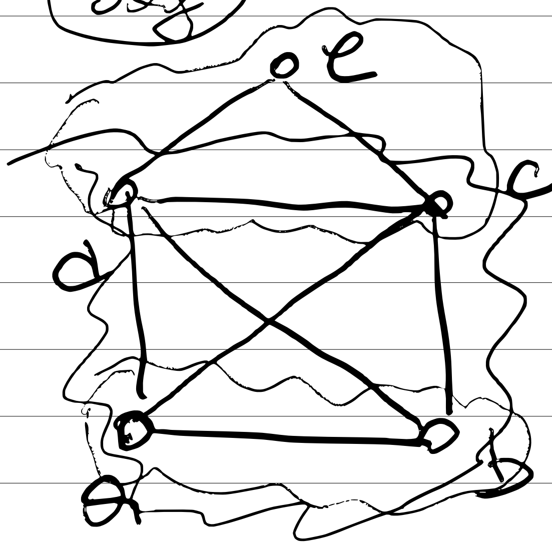
def | Кликово число

Кликово число на граф е

броят на върховете на най-голямата

клика в графа!

Зад



Какво е кликовото число на графа?

4 върха

a b c d

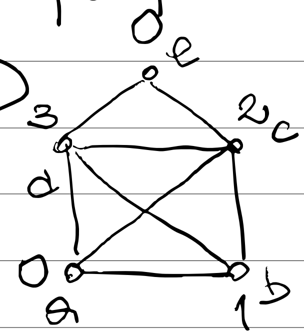
Какъв е броя на най-голямото антиклика в същият граф : {a, e}, {b, e}

2

def | Върхово хроматично число

Това е най-малкият брой цветове необходими за оцветяване на върховете на граф, така че никои два едноцветни върха да не са свързани с ребро.

Зад



Какво е върховото хром. число на графа?

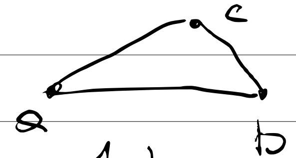
кликово – 4

4 ✓

Th | Върховото хроматично число е по-голямо или равно на кликовото число.

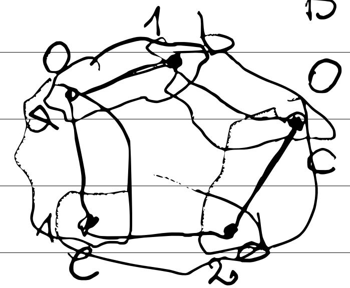
323 Постройте пример за th-ята, за който се получава ">".

C_3:



кликово = 3, върхово = 3.

C_5:



върх. хром. число = 3

кликово = 2

антикликово = 2

Th на Кьониг/

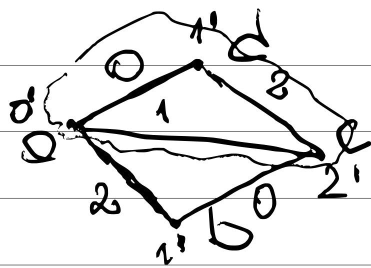
Един граф е двуцветен/двуделен $\Longleftrightarrow$ графът не съдържа цикъл с нечетна дължина.

Да минем към оцветяване на ребра.

def| Реброво хром. число

Минимален брой цветове необходими за оцветяване на ребрата, така че никои две едноцветни ребра да нямат общ връх.

зад



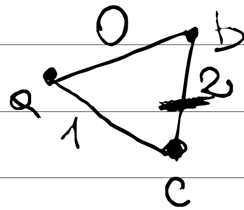
Какво е ребровото хром. число на графа?

$d(a) = d(c) = 3$

клика = 3, върхово хром = 3.

Th| Ребровото хроматично число е $\geq \max\limits_{v \in V} d(v)$.

зад) Да построй граф, за който е изпълнено „<".



реброво хром = 3, а $\max_{v \in V} d(v) = 2$.

Зад. Нека $G = \langle V, E \rangle$ е дърво.

Какви са:

a) кликовото число;

b) върховото хром. число;

c) ребровото хром. число?

→ ацикличен е ⇝ клик. число е макс 2.

↳ $|V| = 0$ ⇝ 0

↳ $|V| = 1$ ⇝ 1

↳ $|V| > 1$ ⇝ 2

→ ацикличен ⇝ няма цикли от нечетна дължина ⇝ двуделен.

↳ $|V| = 0$ ⇝ 0

↳ $|V| = 1$ ⇝ 1

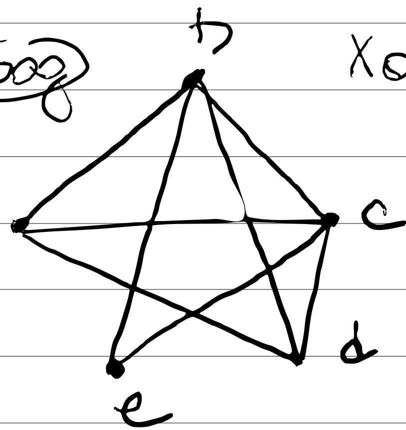
↳ $|V| > 1$ ⇝ 2

→ $\underset{?}{=} \max_{v \in V} d(v)$.

def 1 • Хамилтонов път – минаване точно веднъж през всеки връх на графа (първи ≠ последен).

• Хамилтонов цикъл – хамилтонов път, в който първи = последен.

зад.

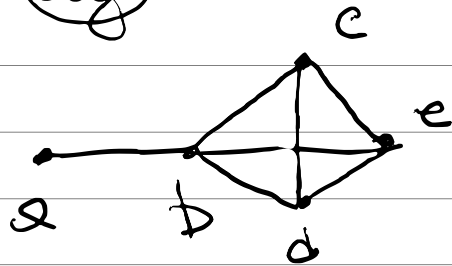


Хамилтонов ли е графа?

Да. Примери:

- adbeca
- ecdabe

зад



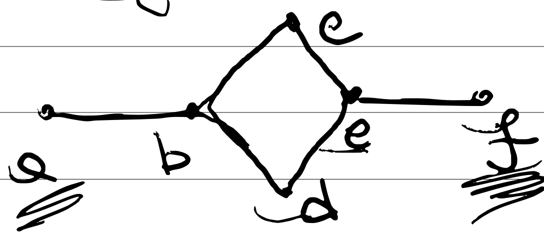
a) Хамилтонов ли е?

b) Има ли Хамил. път?

→ Не: $d(a) = 1$.

→ dceba

зад



a) Хамилтонов ли е?

b) Има ли Хамил. път?

→ Не!

→ Не!

abcef → няма d

abdef → няма c

Не!

КНГ

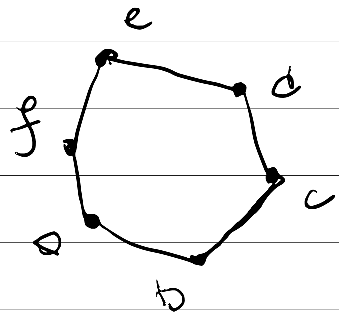
Th/ на Dirac

Ако $G = \langle V, E \rangle$ е граф, т.ч.
$n = |V| \geq 3$ и всеки връх $v \in V$, то
$d(v) \geq \frac{n}{2}$, то G е Хамилтонов.

Th/ на Оре

Ако $G = \langle V, E \rangle$ е граф, т.ч.
$n = |V| \geq 2$ и за всяка двойка
не са свързани с ребро → несъседни върхове u и $v \in V$
е изпълнено, че $d(u) + d(v) \geq n$, то
G е Хамилтонов.

Зад. Условието $d(u) + d(v) \geq n$ е достатъчно, но не е необходимо, за да бъде графа Хамилтонов.



Хамилтонов е

$|V| = 6$

$(\forall v \in V)\left[d(v) = 2 \ngeq \frac{6}{2} = 3\right]$

$(\forall v \in V)(\forall u \in V)\left[\{u, v\} \notin E \Rightarrow d(v) + d(u) = 2 + 2 = 4 \ngeq 6\right]$

(309) На прием били поканени 20 гости. Всеки от тях познава поне 10 от другите. Док, че можем да ги наредим на масата, така че всеки да си познава двамата съседа.

КНГ: V - гостите и $|V| = 20$

E - познанства м/у тях.

$$(\forall v \in V)\left[d(v) \geq 10 = \frac{20}{2}\right]$$

Th Dirac и има такова нареждане.

def| Път – не може да се повтарят върхове и ребра.

Верига – не може да се повтаря ребро, но може да се повтарят върхове.

def| Ойлерови вериги

Минават през всяко ребро точно веднъж. Има два вида: отворени и затворени.

Ойлеров път = отворена верига

Ойлеров цикъл = затворена верига

Th | Нека G е <u>свързан мултиграф</u>.
Ако броят на върховете от нечетна степен е:

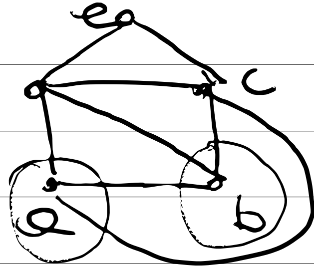
a) 0, то има затворена Ойлерова верига и <u>не същ. отворена.</u>

b) 2, то същ. отворена Ойлерова верига с краищ(а) двата върха от нечетна степен и не същ. затворена;

c) ≠ 0 и 2, то не същ. нито отворена, нито затворена Ойлерова верига.

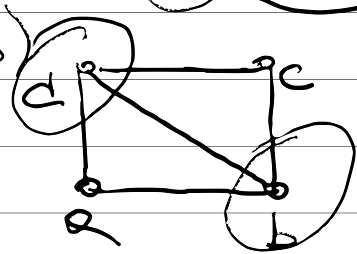
зад 6 Има ли Ойлерова верига? Ако да, то каква е тя?

a)



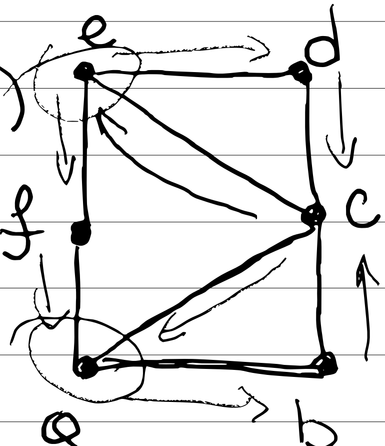
Ойлеров път.

b)



Ойлеров път:
dabdcb

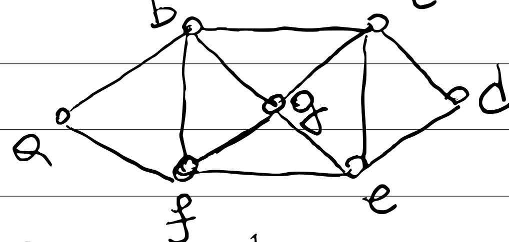
c)



Ойлеров път:
edcabcefa

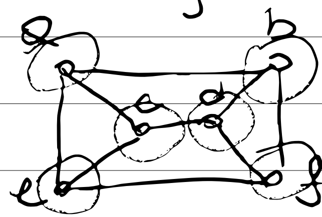
309 Какви са следните графи?

а) Ойлеров ли е?



Затворено верига:

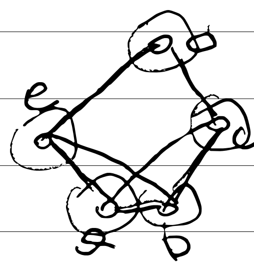
b)



Ойлеров или Хамилтонов? → не е Ойлеров.

По th на Dirac, той е Хамилтонов

c)

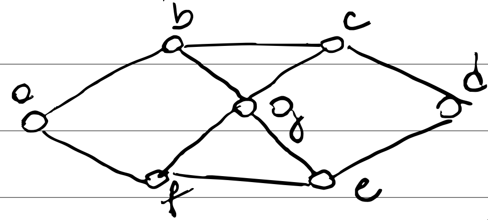


Ойлеров или Хамилтонов? → не!

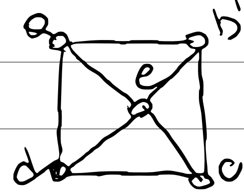
Хамилтонов: $\frac{5}{2} = 2,5$

$d(v) + d(u) \geq 5 \xrightarrow[\text{Оре}]{\text{th на}}$ Хамилтонов

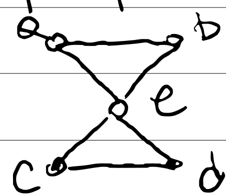
d) Ойлеров и Хамилтонов?



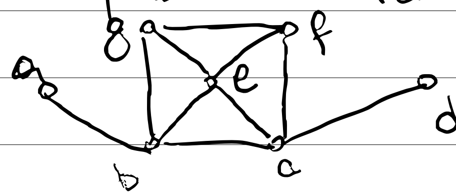
e) Хамилтонов, но не е Ойлеров



f) Ойлеров, но не Хамилтонов.



g) Не Ойлеров и не Хамилтонов



Покриващи дървета (ST), Оптимални покриващи дървета (OST), Покриващи дървета на най-късите пътища (SPST)

def| Покриващо дърво

$T = \langle V, \mathcal{E}' \rangle$ е покриващо дърво на $G = \langle V, \mathcal{E} \rangle$, ако:

- $\mathcal{E}' \subseteq \mathcal{E}$
- G е свързан (иначе T е гора);
- T е дърво.

Ако графът има тегла по ребрата, можем да говорим за Optimal spanning tree (OST) било то максимални или минимални.
Минимално покриващо дърво свързва всички върхове на даден свързан теглови неориент. граф и сбора от теглата на ребрата е минимален.

Основни алгоритми строящи покриващи дървета на НГ/ОГ.
- bfs – в ширина
- dfs – в дълбочина

Имаме теорема за НГ:

Th | Всяко кореново дърво е дърво

Какво и:

Th | Графът G има покриващо дърво $\leftrightarrow$ G е свързан

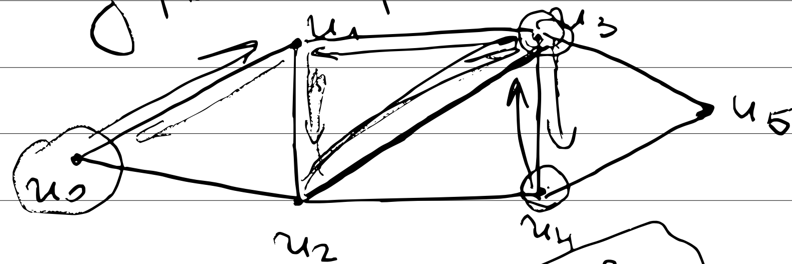
```
BFS(G, s)
    for all v ∈ V:
        dist(v) = ∞
        parent(v) = null
    dist(s) = 0 // тривиален път от s до s
    queue = initQueue(s) // в опашката
    са сиви върхове, а извън нея
    са или бели, или черни. Белите
    имат dist = ∞, а черните dist < ∞ //
    while queue is not empty:
        v = dequeue(queue) // вземане главата на опашката

        for all edges {v, u} ∈ E:
            if dist(u) = ∞: // бял връх
                enqueue(queue, u)
                dist(u) = dist(v)+1
                parent(u) = v
```

масив - пази разстояние от s до останалите

масив пази покриващото дърво

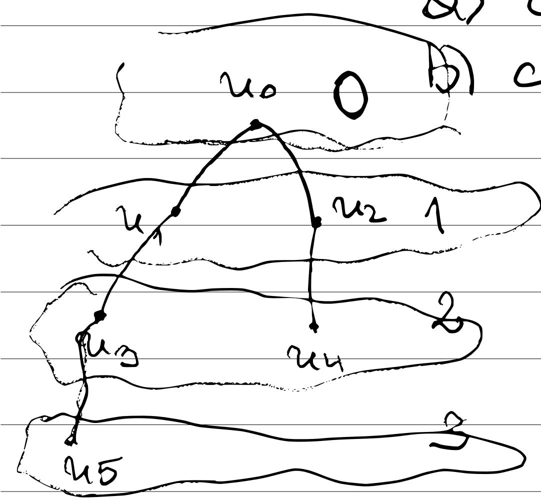
```
DFS(G)
  forall v ∈ V:
    visited(v) = false   // сочи бели
    parent(v) = null
  forall v ∈ V
    if not visited(v):
      explore(G, v)
explore(G, v)
  visited(v) = true   // сив връх е v
  for all edges {v, u} ∈ E:
    if not visited(u):
      parent(u) = v
      explore(u)
  // вече тук v е черен.
```

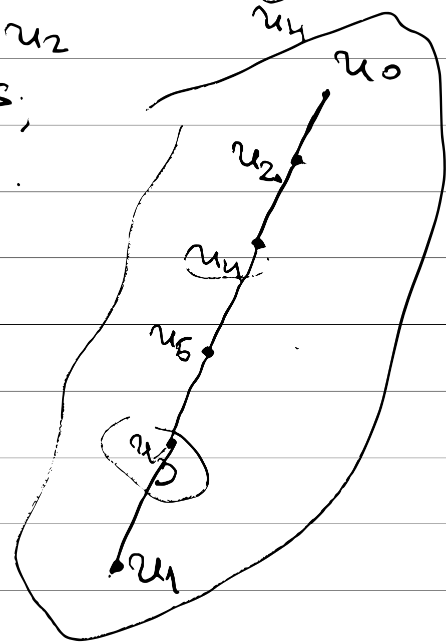
(302) Да се номери покриващо дърво на графа!



а) с bfs;

б) с dfs.



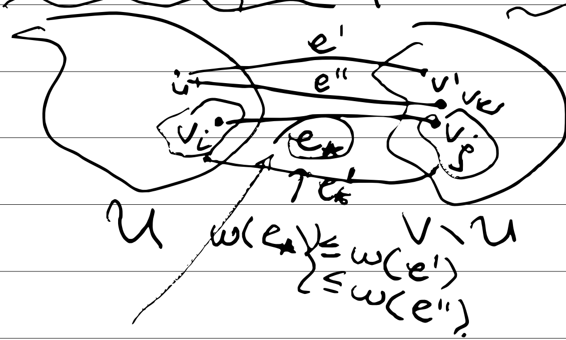


Optimal spanning tree

```
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 $G=\langle V, E\rangle$ 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. $e \in E_0$.



$e_0 < e_1 < e_2$ - ——

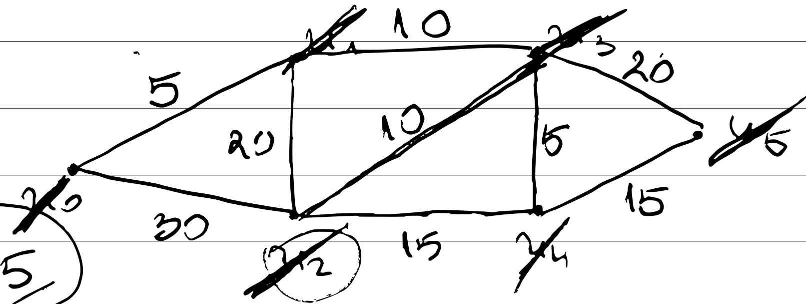
Prim-Jarnik(G, w, s)

```
Prim-Jarnik(G, w, s)
  for all 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
  for each {u,v} ∈ E:
      if cost(v) > w(u,v):
          cost(v) = w(u,v)
          parent(v) = u
          decreaseKey(PQ, v)
```

намерили сме връх, за който можем да подобрим цената му

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.



$u_3 u_4 - 5$

$u_0 u_1 - 5$

$u_1 u_3 - 10$

$u_2 u_3 - 10$

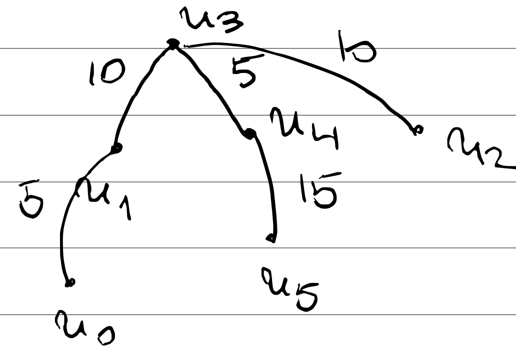
$u_2 u_4 - 15$

$u_4 u_5 - 15$

$u_3 u_5 - 20$

$u_1 u_2 - 20$

$u_0 u_2 - 30$



SPST

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

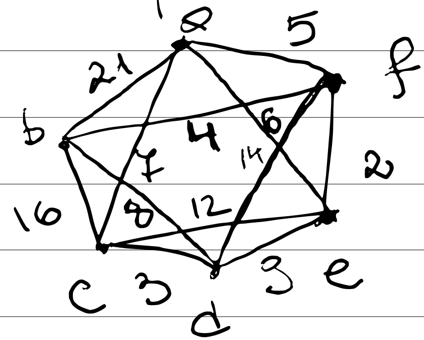
* Ако сред теглата има отрицателни, има други алгоритми, които строят SPT.
* Dijkstra строи дърво на най-късите пътища от даден връх, до всички други (работи за ДОГ и КМР).

! $MST \neq SPST$!

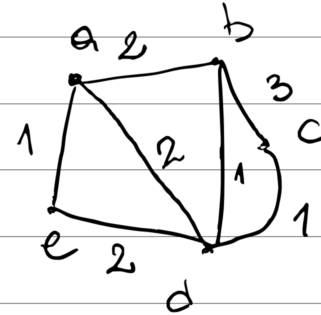
Тв | Всеки начален отрез на най-къс път от s строен от Dijkstra е също най-къс път.

```
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)        взимаме върха, до който стигаме най-бързо
    for all edges {u,v} ∈ E:
      if dist(v) > dist(u) + w(u,v)        намерили сме по-къс път до v през u
        dist(v) = dist(u) + w(u,v)
        parent(v) = u
        decreaseKey(PQ, v)
```

329 Постройте MST на графе с начало a.



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



$$M = \begin{pmatrix} \infty & 2 & \infty & 2 & 1 \\ 2 & \infty & 3 & 1 & \infty \\ \infty & 3 & \infty & 1 & \infty \\ 2 & 1 & 1 & \infty & 2 \\ 1 & \infty & \infty & 2 & \infty \end{pmatrix}$$

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.