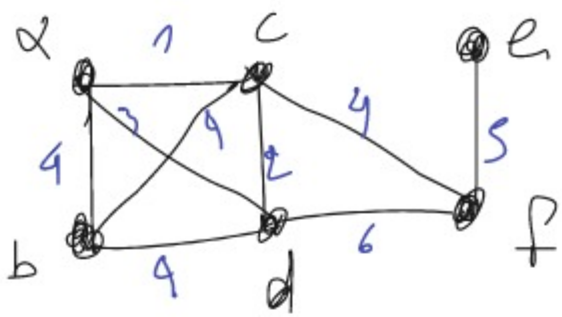
Ελάχιστο επικαλύπτον δέντρο



Είσοδος: $G = (V, E)$

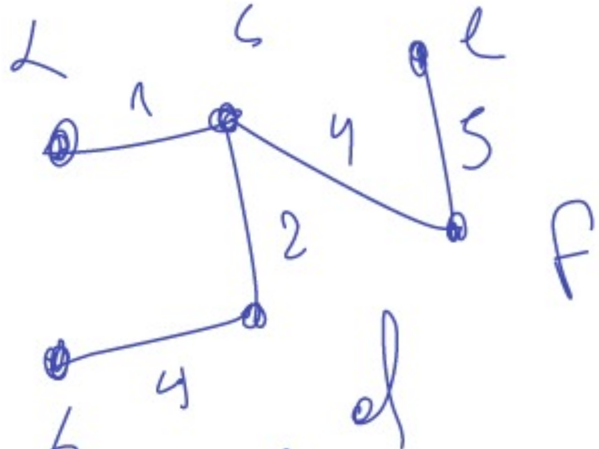
$w : E \to \mathbb{N}$

Έξοδος: MST (δέντρο

$T = (V, E')$, $E' \subseteq E$

με το ελάχιστο

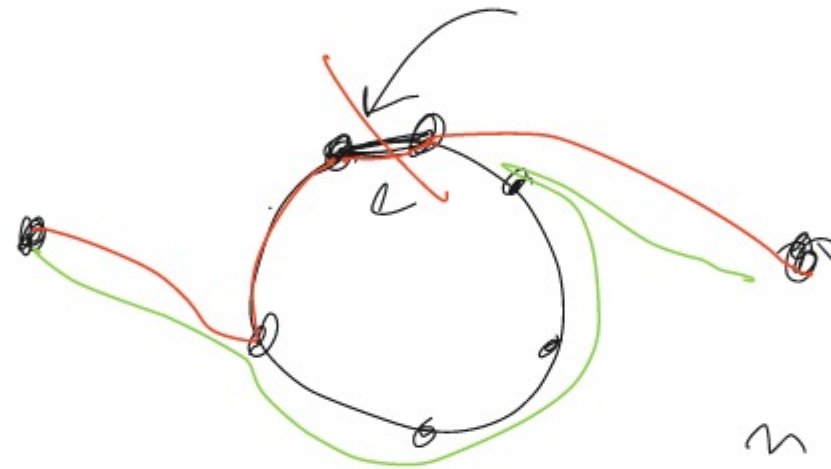
$weight(T) = \sum_{e \in E'} w(e)$)



MST

① Η αφαίρεση μιας ακμής κύκλου δεν βλάπτει την συνεκτικότητα.

Απόδ.

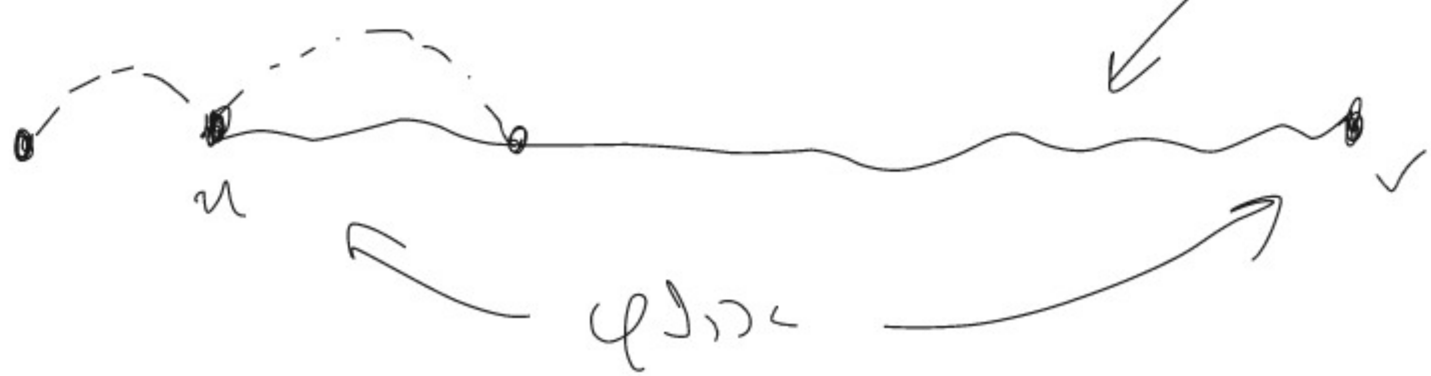


$n = |V|$, $m = |E|$

② Κάθε δένδρο με $n \geq 2$ έχει ≥ 2 φύλλα

Απόδ.

μέγιστο μήκος



③ Δέντρο με n-κορυφές $\iff$ συνεκτικό με ακριβώς $(n-1)$-ακμές

<u>Απόδ.</u>

($\Rightarrow$) Επαγωγή στο n.

<u>Βάση:</u> $n=1$, ισχύει

<u>Ε.Υ.:</u> Δέντρο n-κορυφές $\Rightarrow$ $(n-1)$-ακμές.

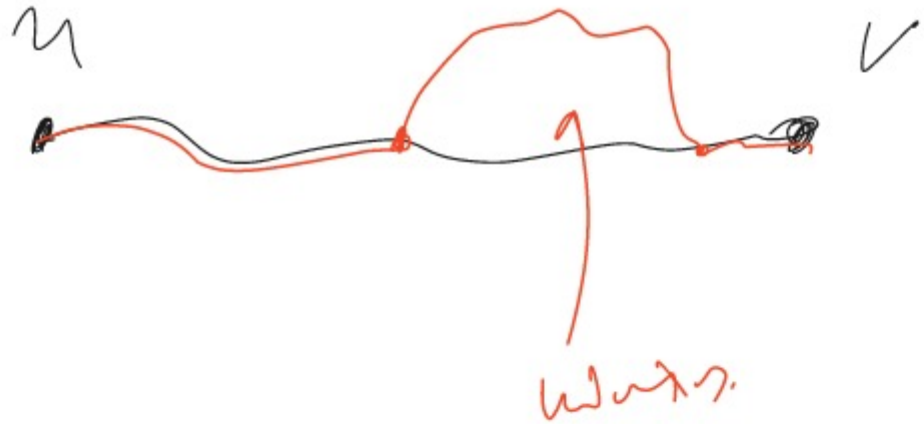
<u>Ε. Βήμα:</u> Έστω δέντρο T με $n+1$ κορυφές.

② $\Rightarrow$ Έστω u φύλλο του T ...

(⇐) Έστω (προς άτοπο) συνεκτικό γράφημα G με n κορυφές και (n-1)-ακμές που περιέχει κύκλο C. Αφαιρούμε ακμές κύκλων όσο υπάρχουν κύκλοι. Από το 1 το γράφημα που προκύπτει από διαγραφές είναι συνεκτικό, δεν έχει κύκλο άρα έχει ακριβώς n-κορυφές. Από την (⇒) το G' έχει (n-1)-ακμές. Άτοπο.

④ Δέντρο $\Leftrightarrow$ κάθε 2 κορυφές συνδέονται με
<u>μοναδικό</u> μονοπάτι.

<u>Απόδ.</u>
($\Rightarrow$)



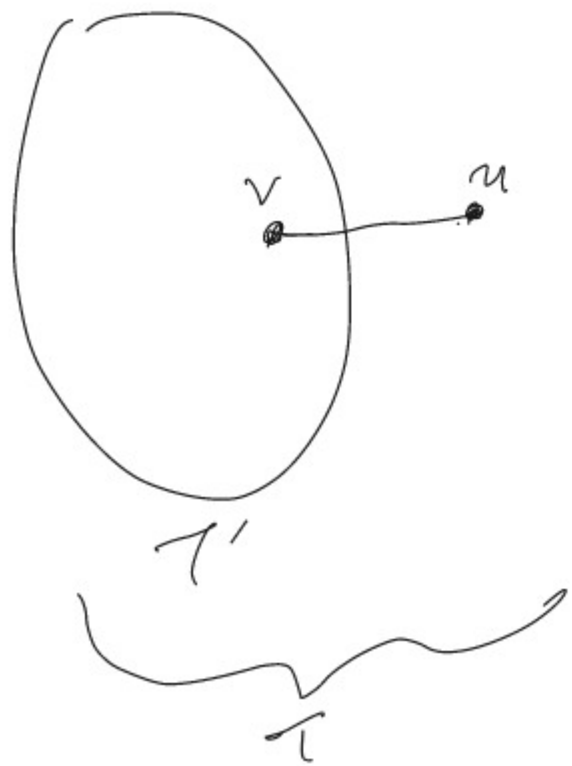
Επαγωγή στο n. (πλήθος κορυφών δέντρου)

<u>Βάση</u>: $n=1$ ισχύει

<u>Ε.Υ.</u>: Σε κάθε δέντρο με n-κορυφές υπάρχει μοναδικό μονοπάτι.

miro

Ε. Βήμα: Έστω δέντρο T με $(n+1)$-κορυφές. Έστω u φύλλο του T. Έστω T' το δέντρο που προκύπτει από την αφαίρεση του u. Το T' έχει n-κορυφές!

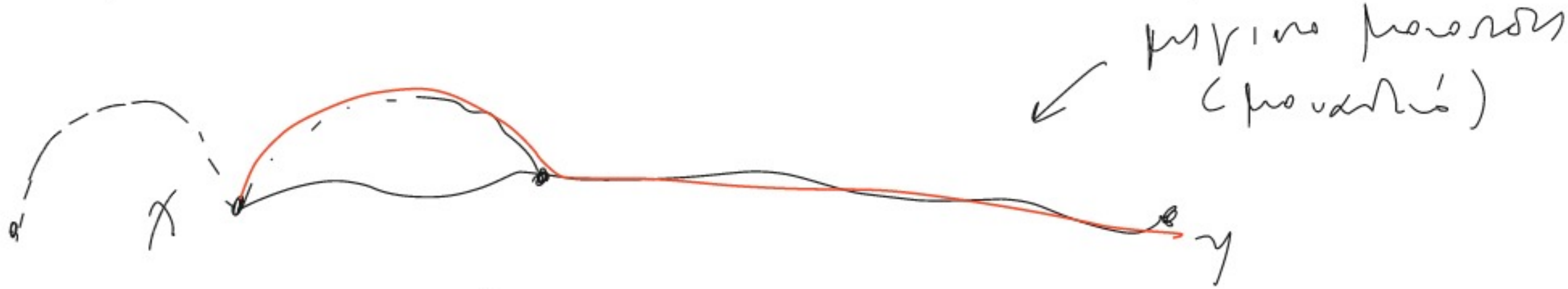


(⇐) Επαγωγή στο n;

<u>Βάση:</u> n=1

<u>Ε.Υ.:</u> κάθε γράφημα με n-κορυφές που ικανοποιεί με μοναδικό μονοπάτι είναι δέντρο.

<u>Ε. Βήμα:</u> Έστω G με (n+1)-κορυφές που —//— .



Η x έχει βαθμό 1 την αφαιρώ και μένει το G' το οποίο έχει n-κορυφές και μοναδικό μονοπάτι άρα και το G' ισχύει η Ε.Υ. δηλαδή είναι δέντρο.

Άρα και το G είναι δέντρο.

Αλγόριθμος Kruskal:

kruskal(G,w):

Είσοδος: $G=(V,E)$, $w: E \to \mathbb{N}$

Έξοδος: M.S.T. (το ορίζουν οι ακμές του X)

for every $u \in V$
    makeset(u) $\leftarrow O(|V|)$

$X = \emptyset$

Ταξινόμησε τις ακμές του E κατά αύξουσα σειρά βάρους $\leftarrow O(|E| \cdot \log|V|)$

for every $\{u,v\} \in E$ (κατά αύξουσα σειρά βάρους)
    if find(u) $\neq$ find(v)
        $X \leftarrow X \cup \{\{u,v\}\}$
        union(u,v)

$O(|E| \cdot \log|V|)$

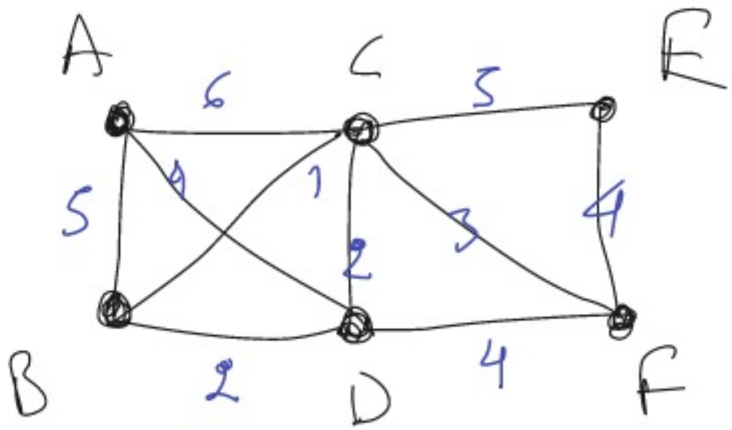
Τελικώς $O(|E| \log|V|)$

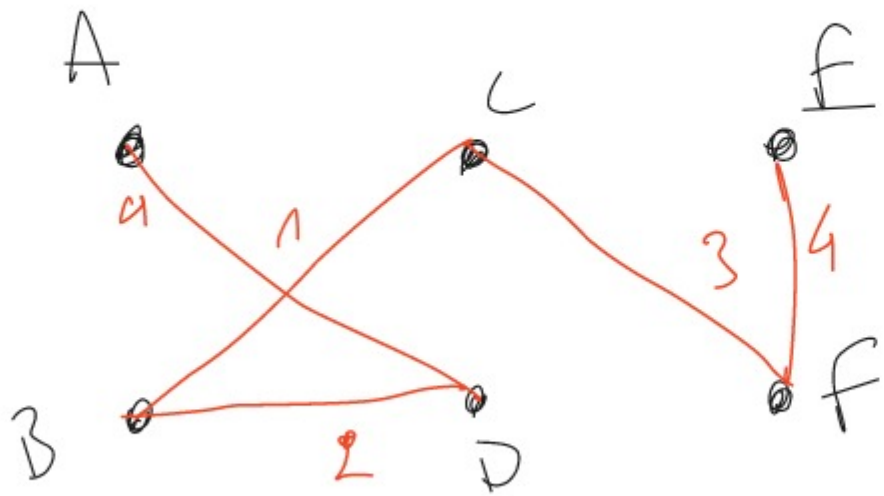
---

makeset(x) → Φτιάχνει το $\{x\}$ $\leftarrow O(1)$

find(x) → βρίσκει το σύνολο που ανήκει $\leftarrow O(\log n)$

union(x,y) → ενώνει τα σύνολα που ανήκουν τα x και y $\leftarrow O(\log n)$

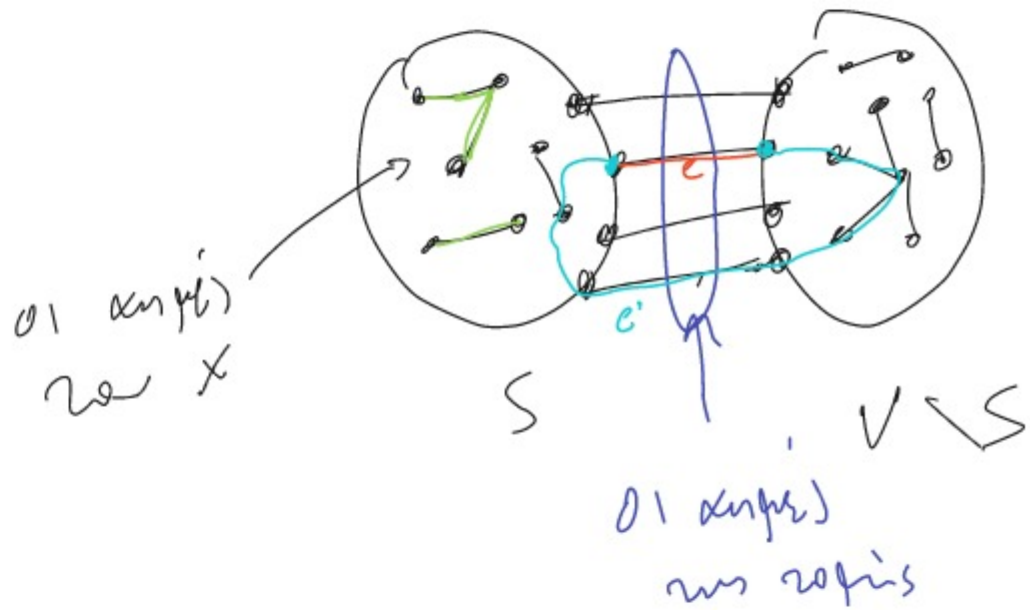




⇝ bipos

Ιδιότητα τομής:

Έστω ότι οι ακμές του X ανήκουν σε ένα MST. Επιλέγουμε $S \subseteq V$ τ.ω.:



$w(e) \leq w(e')$

Έστω e η ακμή με το μικρότερο βάρος από τις ακμές της τομής. Τότε το $X \cup \{e\}$ είναι τμήμα ενός MST.

X ανήκει σε ένα MST έστω το $T = (V, E_1)$

- $e \in E_1$: προφανώς $X \cup \{e\}$ ανήκει στο T.
- $e \notin E_1$. Έστω $T' = (V, (E_1 \setminus \{e'\}) \cup \{e\})$

Δέντρο καθώς είναι ακυκλικό και έχει $|V|-1$ ακμές.

$$w(T) = \sum_{e \in T} w(e)$$

$$w(T') = W(T) - w(e') + w(e) \Rightarrow$$

$$\Rightarrow w(T') \leq w(T) \xRightarrow{T \text{ MST}}$$

$$\Rightarrow w(T') = w(T) \Rightarrow T' \text{ MST}.$$

Αλγόριθμος Prim!

Prim (G, w):

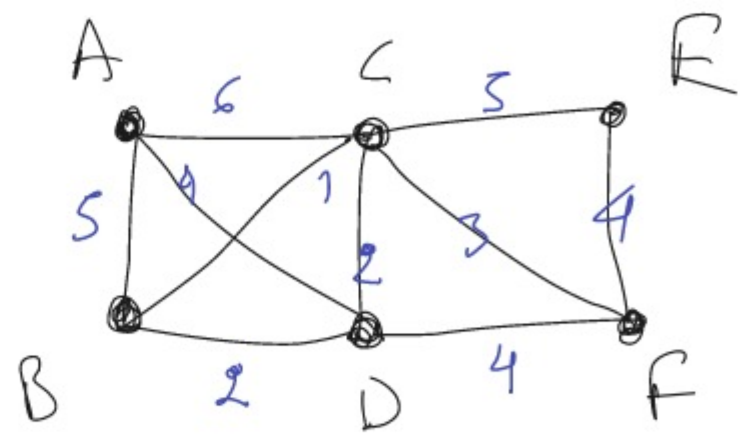
Είσοδος: ——//——

Έξοδος: ——//—— (το δέντρο ορίζεται από την συστοιχία prev)

$$\left.\begin{array}{l}\text{for every } u \in V \\ \quad cost(u) = \infty \\ \quad prev(u) = NIL \\ \text{choose } u_0 \in V \\ \quad cost(u_0) = 0\end{array}\right\} O(|V|)$$

H = make queue (V) (ουρά προτεραιότητας, με τα κόστη για προτεραιότητα) $\leftarrow O(|V|)$

$$\left.\begin{array}{l}\text{while } H \neq \emptyset \\ \quad v = \text{deletemin}(H) \quad \leftarrow O(\log|V|) \\ \quad \text{for every } \{v, z\} \in E: \\ \quad\quad \left.\begin{array}{l}\text{if } cost(z) > w(v,z) \\ \quad cost(z) = w(v,z)\end{array}\right\} O(1) \\ \quad\quad\quad prev(z) = v \\ \quad\quad\quad \text{decrease key}(H, z). \quad \leftarrow O(\log|V|)\end{array}\right\} O(|E| \cdot \log|V|)$$



вершины в MST	A	B	C	D	E	F
∅	0/NIL	∞/NIL	∞/NIL	∞/NIL	∞/NIL	∞/NIL
{A}		5/A	6/A	4/A	-//-	-//-
{A, D}		2/D	2/D		-//-	4/D
{A, D, B}			1/B		-//-	4/D
{A, D, B, C}					5/C	3/C
{A, B, C, F}					4/F	