Δένδρα επικάλυψης ελάχιστου κόστους
και
το πρόβλημα του πλανόδιου πωλητή
(Traveling Salesman Problem: TSP)

Αλγόριθμος Prim

- Ξεκινάμε από ένα δένδρο Τ αποτελούμενο από **ένα μόνο** κόμβο.

- Στη συνέχεια, σε κάθε επανάληψη, αυξάνουμε το δένδρο Τ συνδέοντάς το στο **πλησιέστερο ελεύθερο κόμβο** ως προς την έννοια του βάρους (κόστους).

Αλγόριθμος Prim

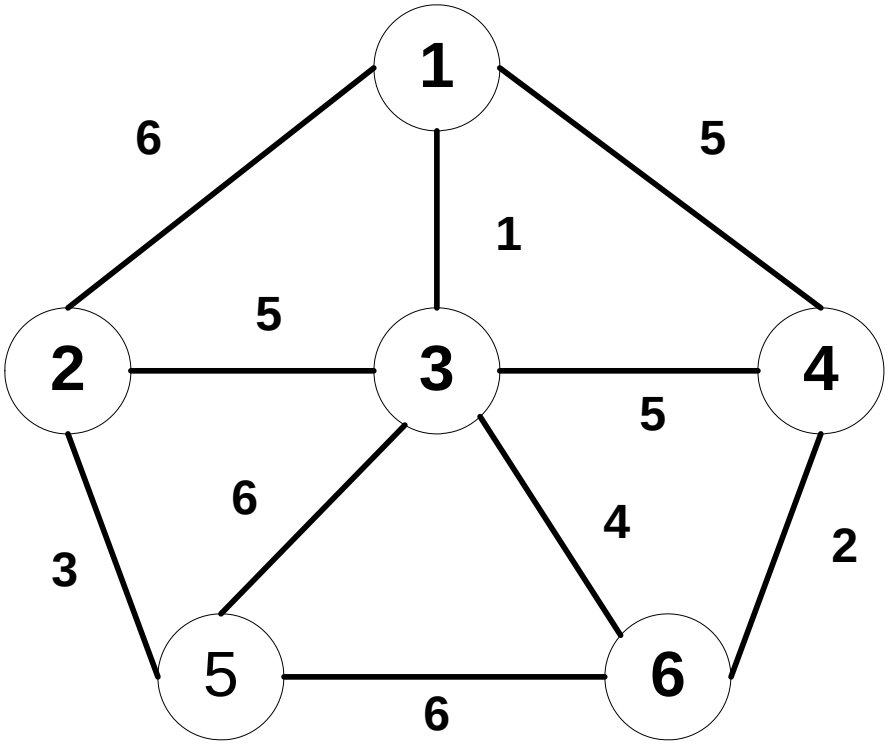
G=(V,E)

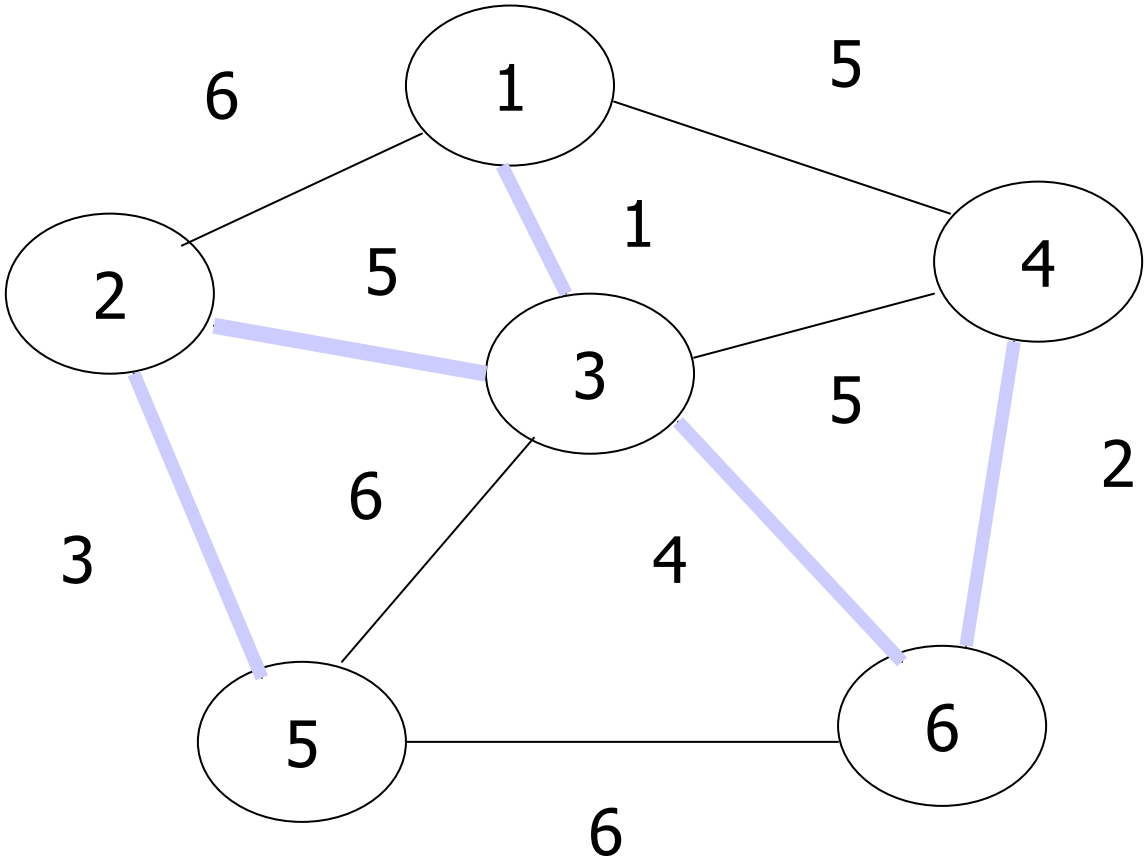
Τ={u, οποιοσδήποτε κόμβος}

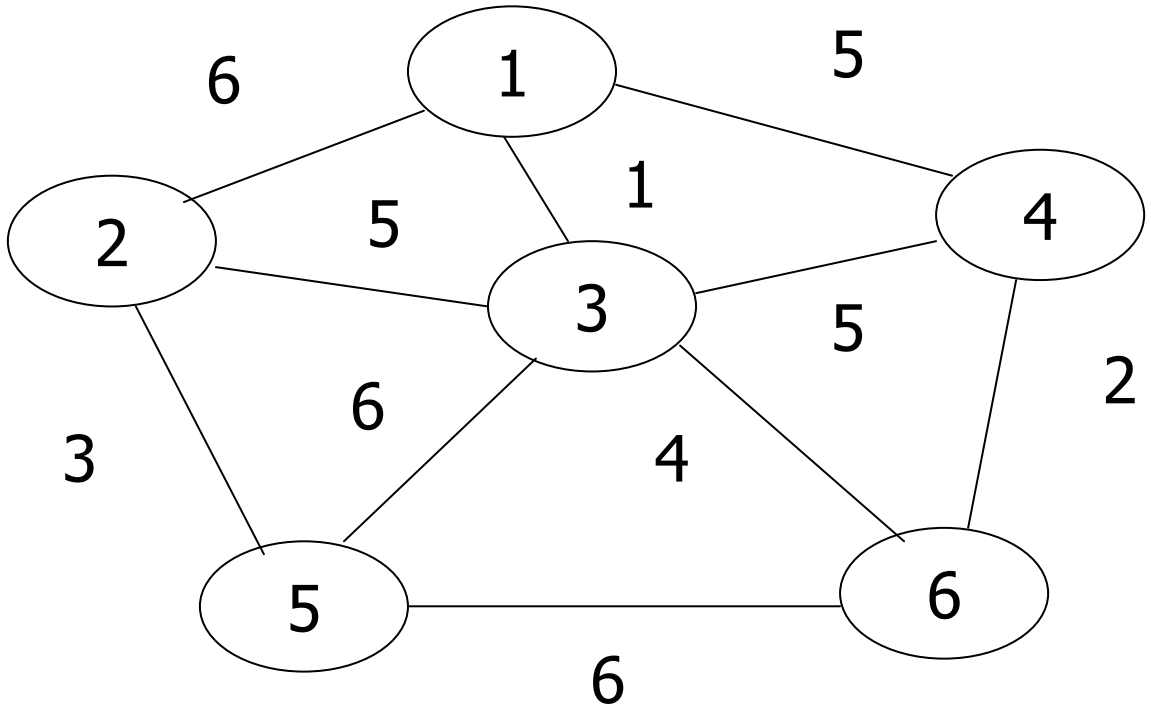
Repeat

Τ←Τ ∪ {v}, v πλησιέστερος ελεύθερος κόμβος στο Τ

Παράδειγμα

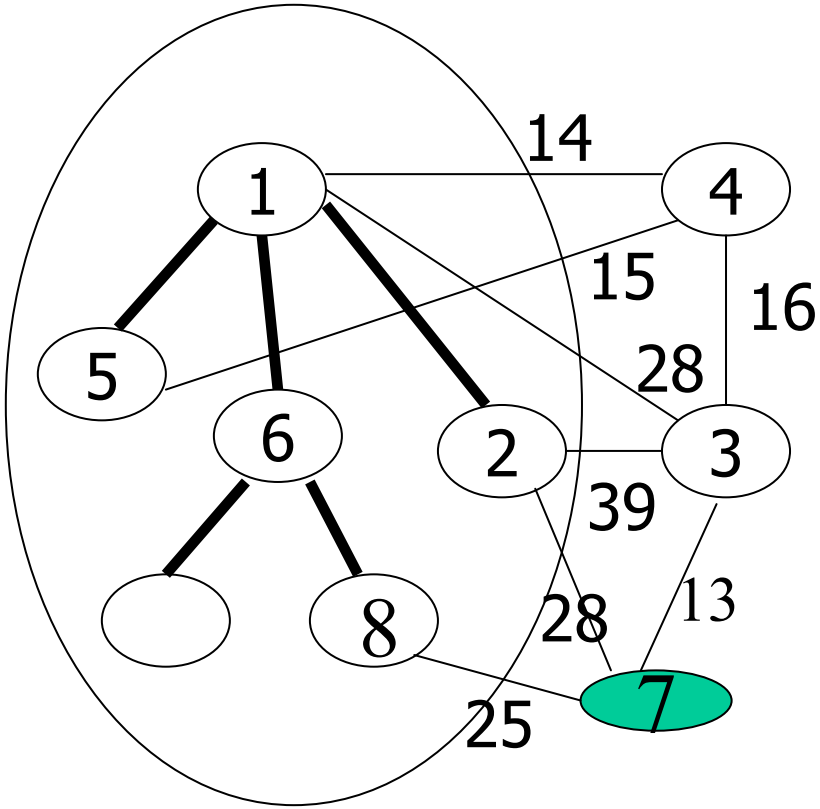



1
2
3
4
5
6
6
5
1
5
5
2
6
3
4
6



near[x]= { x, αν x **ανήκει** στο T
0 αν [x,y] **δεν ανήκει** στο E, για όλα τα y του T
y, αν y **ανήκει** στο T (ο πλησιέστερος κόμβος)

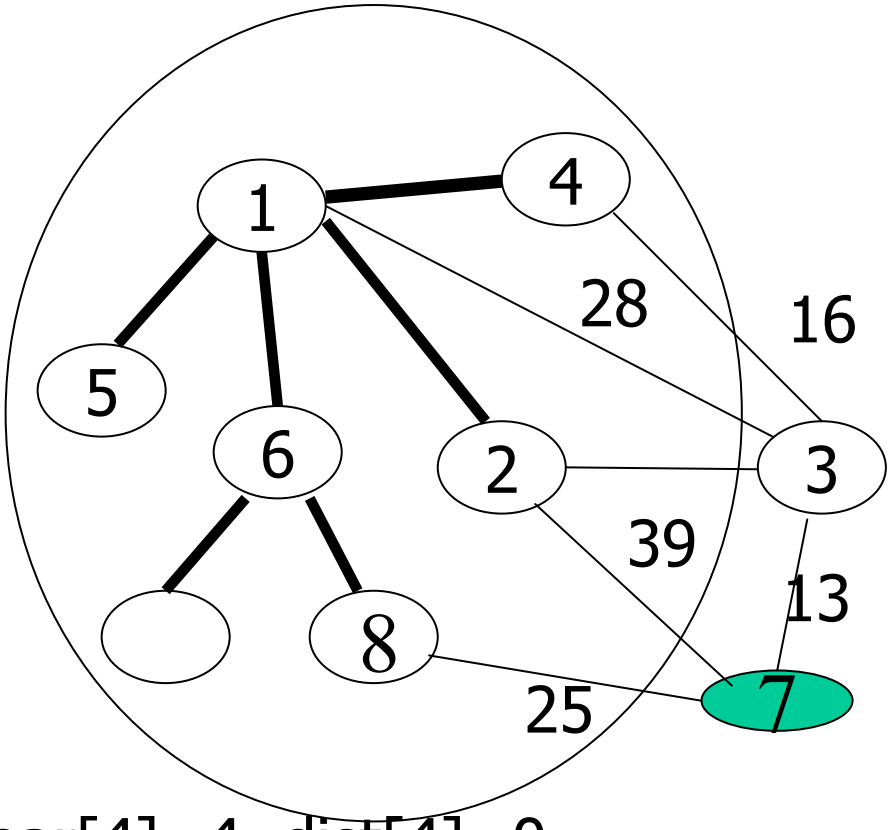
dist[x]= { $W_{x,y}$ αν near[x]=y
$+\infty$ αν near[x]=0
0 αν near[x]=x



near[4]=1, dist[4]=14

near[3]=1, dist[3]=28

near[7]=8, dist[7]=25



near[4]=4, dist[4]=0

near[3]=4, dist[3]=16

near[7]=8, dist[7]=25

Αλγόριθμος Prim (nearest neighbour)

```
initialization;

choose  s   arbitrary; near[s] ← s; dist(s)=0;

For i in Γ(s)do
      near(i) ← s;
      dist(i) ← w_{s,i};

for every node i other than s and not in Γ(s)do
      begin
        near(i) ← 0; dist(i) ←  ∞;
      end;
      V_T ← {s}; E_T ← ∅;
```

Αλγόριθμος Prim (nearest neighbour)

```
while |V_T| < n do
  begin
    u ← vertex in(V −V_T) with smallest dist(u);

    if (dist(u) >= ∞ then
       "graph disconnected"; exit;

    E_T ← E_T ∪ {(u, near(u))}; V_T ← V_T ∪ {u},
                          dist(u)=0 near(u)=u;

    for x in Γ(u) and x in (V - V_T) do
      if w_ux < dist(x) then
         dist(x) ← w_ux; near(x) ← u

end;   (****)
```

Συνεκτικότητα

$\Rightarrow$ Αν G συνεκτικός $\longrightarrow$ εύρεση του ΔΕΕΚ σε n-1 επαναλήψεις.

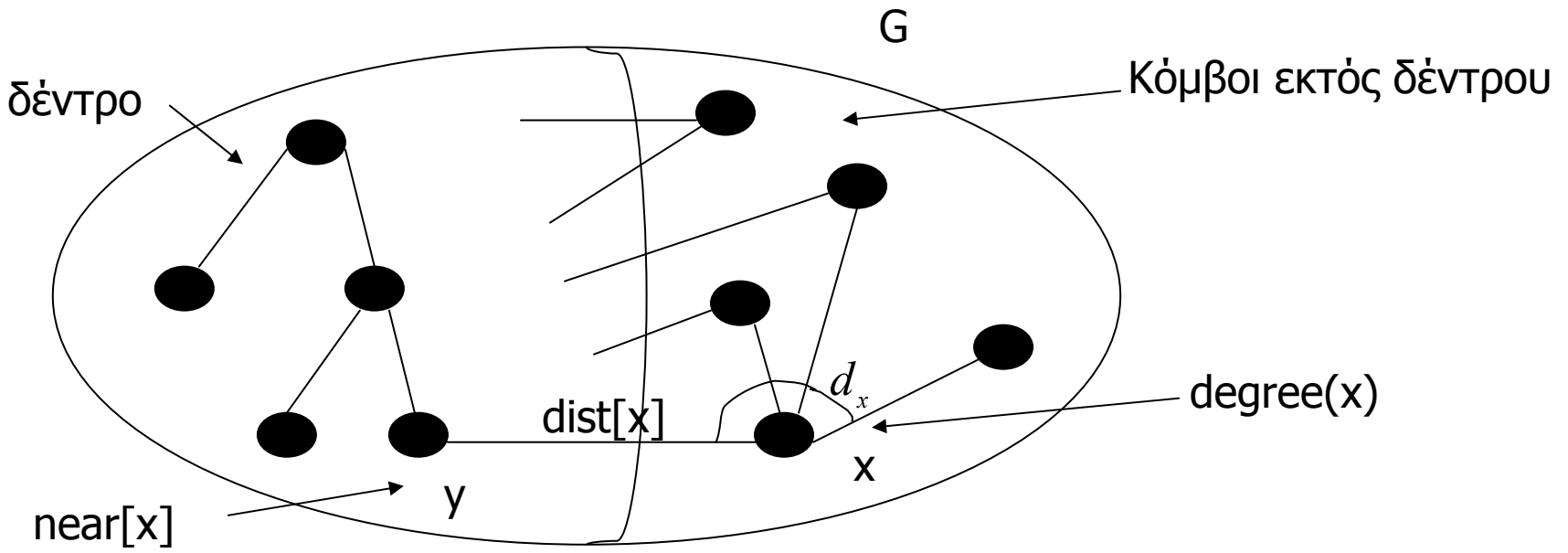
$\Rightarrow$ Αν στην i επανάληψη (i < n-1) δεν ευρίσκουμε πλευρά [x,y] με x στο δένδρο και y έξω από το δένδρο, τότε ο γράφος δεν είναι συνδεδεμένος (αλλά ήδη γνωρίζουμε έλεγχο συνεκτικότητας σε O(m) με την BFS)

Δάσος επικάλυψης

$\Rightarrow$ Αν ο γράφος δεν είναι συνεκτικός, τότε ... ένα δάσος επικάλυψης ελαχίστου κόστους

$\Rightarrow$ Ο αλγόριθμος δίνει τη βέλτιστη λύση (από το θεώρημα βέλτιστου ΔΕΕΚ παρατηρώντας ότι το δάσος εκκίνησης συνίσταται από n μεμονωμένους κόμβους).

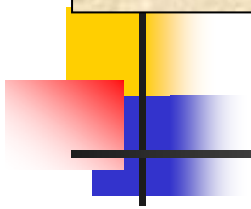
Πολυπλοκότητα Prim



Ο αλγόριθμος σε κάθε επανάληψη χρειάζεται O(n + d_i) πράξεις, όπου d_i ο βαθμός του εισερχόμενου στο δέντρο κόμβου στην i επανάληψη.

Αφού ο αλγόριθμος απαιτεί n-1 επαναλήψεις, η πολυπλοκότητα είναι O(n^2)

Αλγόριθμος Kruskal

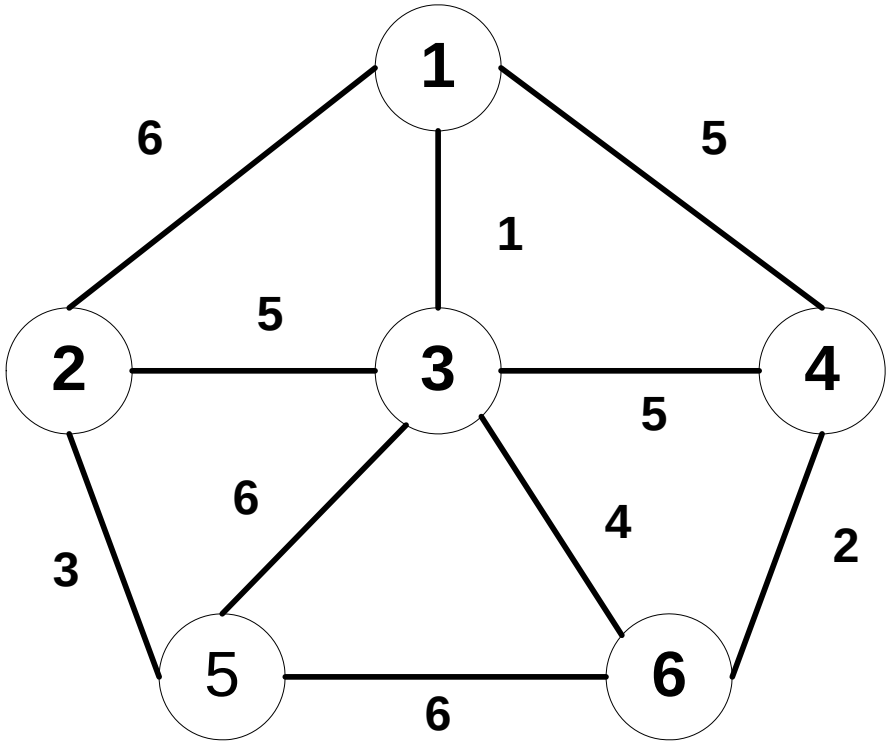


- Ξεκινάμε από ένα δάσος από n δένδρα, κάθε ένα δένδρο εκφυλισμένο σε ένα μεμονωμένο κόμβο.

- Σε κάθε επανάληψη, προσθέτουμε τη πλευρά με το μικρότερο κόστος και η οποία **δεν δημιουργεί κύκλο** με τις ήδη επιλεγμένες πλευρές.

- Σταματάμε όταν έχουμε ένα δένδρο επικάλυψης (γράφος συνεκτικός) ή όταν δεν ευρίσκουμε πλέον πλευρά να προσθέσουμε (ο γράφος δεν είναι συνεκτικός)

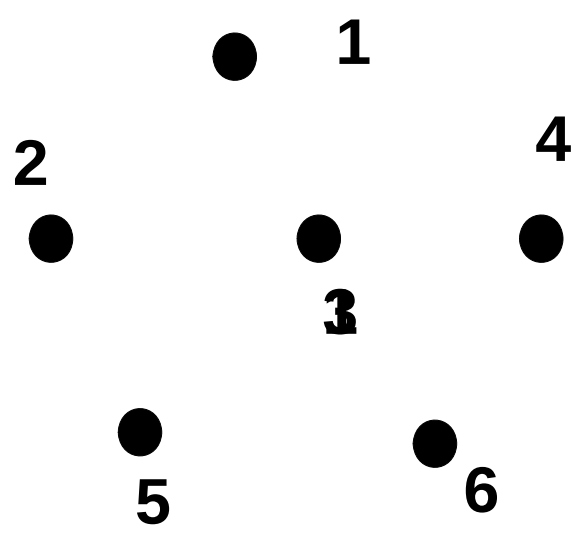
Αλγόριθμος Kruskal - Παράδειγμα



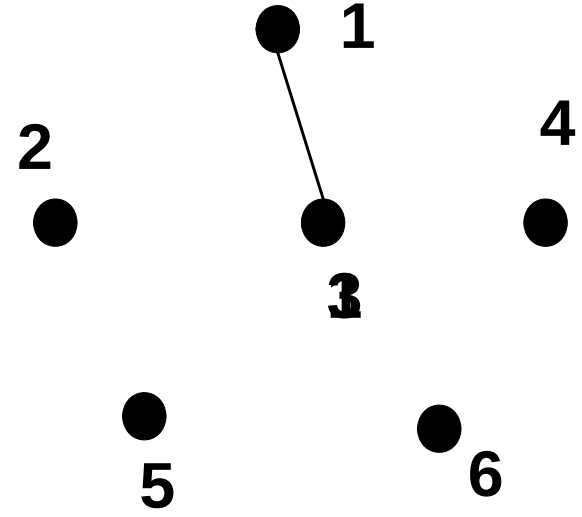
Διάταξη πλευρών κατά αύξουσα τάξη

[1,3] : κόστος 1
[4,6] : κόστος 2
[2,5] : κόστος 3
[3,6] : κόστος 4
[1,4] : κόστος 5
[3,4] : κόστος 5
[2,3] : κόστος 5
[1,2] : κόστος 6
[3,5] : κόστος 6
[5,6] : κόστος 6

Αριθμός συνιστωσών n =
αριθμός μεμονομένων κόμβων

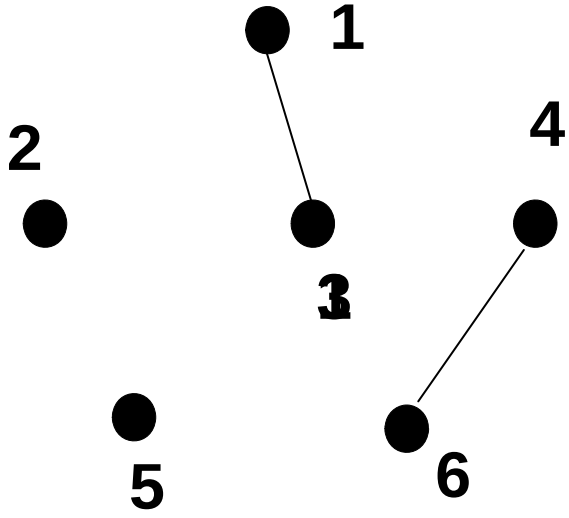


Βήμα 1

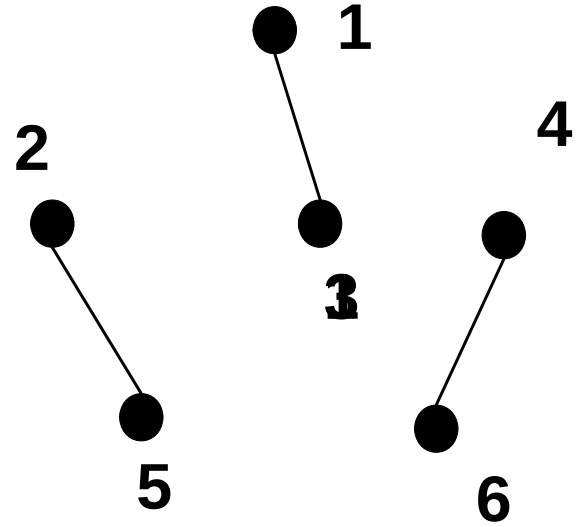


Βήμα 2

Αλγόριθμος Kruskal

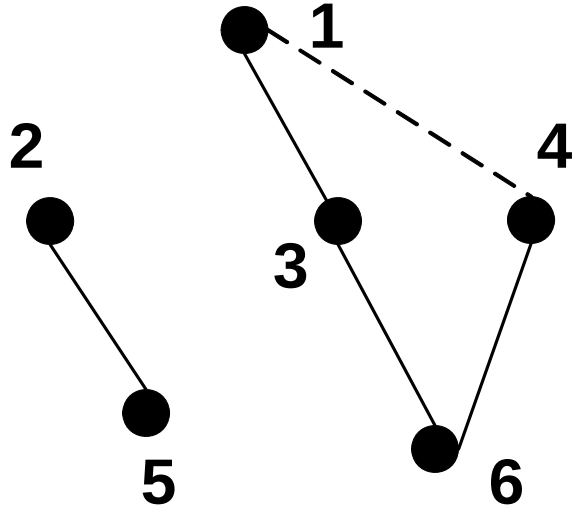


Βήμα 3

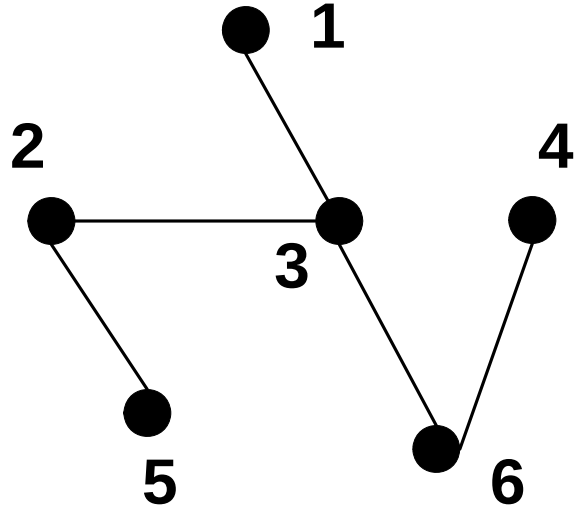


Βήμα 4

Αλγόριθμος Kruskal



βήμα 5



βήμα6

Αλγόριθμος Kruskal

```
T ← ∅
while (|T| < n-1) and (E ≠ ∅) do
    begin
      e ← smallest edge in E;
      E ← E - {e};
      if (T ∪ {e} has no cycle) then
         T ← T ∪ {e}
    end;
if (|T| < n-1) then
   write "network disconnected"; (****)
```

Πολυπλκότητα: O(mlogn)

Εξακρίβωση σχηματισμού κύκλου σε O(logn)

Αλγόριθμος Kruskal - Δάσος Επικάλυψης

- Αν ο γράφος είναι μη συνεκτικός, για να βρούμε ένα δάσος επικάλυψης με τον αλγόριθμο Prim θα πρέπει να επαναλάβουμε τον αλγόριθμο σε κάθε συνεκτική συνιστώσα

- Ο αλγόριθμος Kruskal ευρίσκει απ' ευθείας ένα δάσος επικάλυψης ελάχιστου κόστους.

Αλγόριθμος Kruskal - Πολυπλοκότητα

Ταξινόμηση πλευρών: O(mlog m)

(m αριθμός πλευρών)

Εξακρίβωση σχηματισμού κύκλου: O(logn)

Επαναλήψεις: m

$O(mlogm) + O(mlogn) = O(mlogn)$

δεδομένου ότι $m < n^2$

Σύγκριση των 2 αλγορίθμων

$$\text{Prim} : O(n^2)$$

$$\text{Kruskal} = O(m\log n)$$

$$\pi\upsilon\kappa\nu o i \rightarrow m = O(n^2)$$

$$\Downarrow$$

$$\text{Prim} \rightarrow O(n^2) < O(n^2 \log n)$$

$$\underline{\text{Αραιοί}} \rightarrow m = O(n)$$

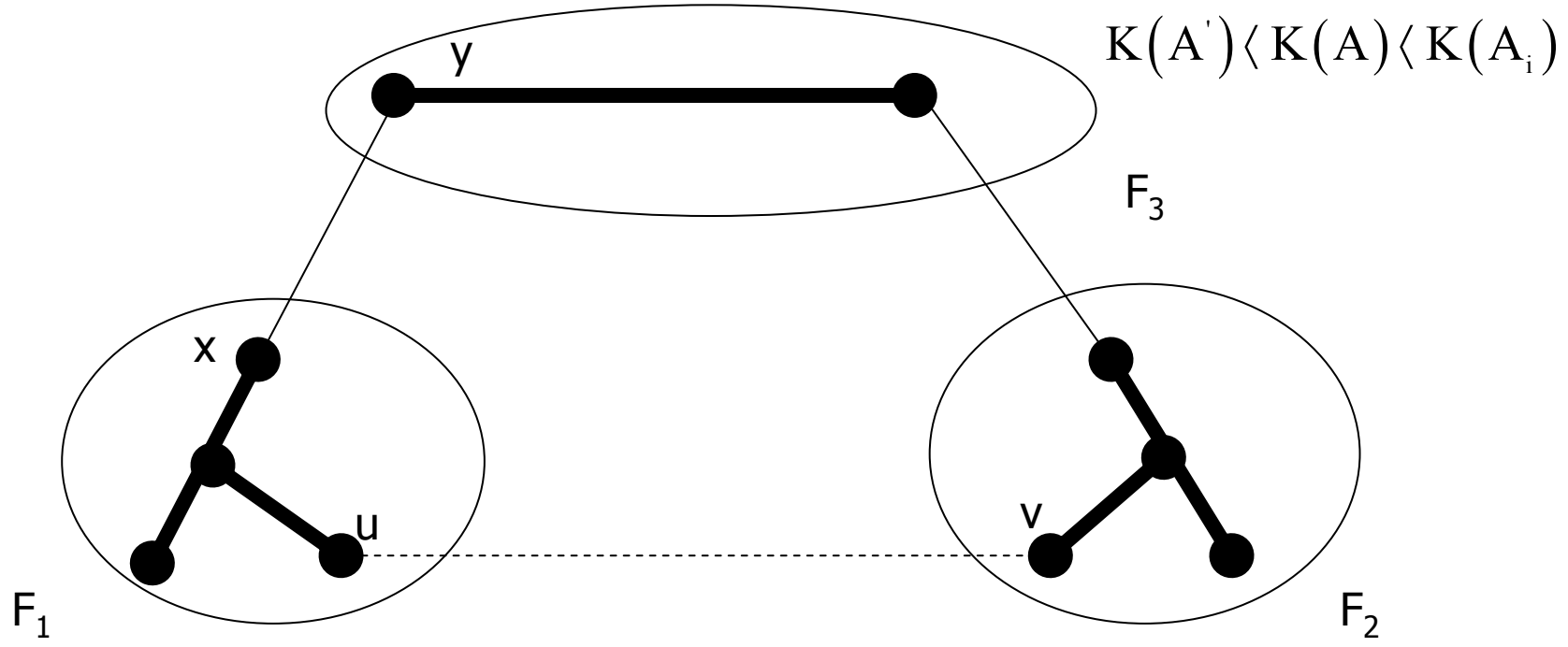
$$\underline{O(n^2) > O(n\log n) \rightarrow \text{Kruskal}}$$

Θεώρημα βέλτιστου για το ΔΕΕΚ

Έστω $F=(F_1, F_2, ...F_k)$ $με$ $F_i = (V_i, E_i)$ ένα δάσος του γράφου $G = (V, E)$ και $[u, v]$ η πλευρά με το ελάχιστο κόστος έχουσα ένα άκρο στο V_1 (G *συνεκτικὸς*). Τότε, ανάμεσα σε όλα τα δένδρα επικάλυψης που περιέχουν αυτό το δάσος, υπάρχει ένα, το καλύτερο το οποίο περιέχει την πλευρά $[u, v]$.

Θεώρημα βέλτιστου για το ΔΕΕΚ - Απόδειξη

$$\text{`Εστω } T = \bigcup_{i=1}^{k} E_i \text{ οι πλευρές του δάσους } F.$$



Απόδειξη (συνέχεια)

Έστω A_i ένα οποιοδήποτε δένδρο επικάλυψης περιέχον το σύνολο των πλευρών Τ.

Ας υποθέσουμε ότι υπάρχει ένα δένδρο επικάλυψης $A = (V, E)$ περιέχον Τ αλλά όχι την πλευρά $[u, v]$ και του οποίου το κόστος είναι ελάχιστο,

δηλαδή $\kappa\acute{o}\sigma\tau o\varsigma\ (A) < \kappa\acute{o}\sigma\tau o\varsigma\ A_i$

Αν προσθέσουμε τη πλευρά $[u, v]$ μέσα στο Ε δημιουργούμε ένα κύκλο c. Υπάρχει λοιπόν πάνω στον κύκλο c μία πλευρά $[x, y]$ τέτοια ώστε $x \in V_1$ $\kappa\alpha\iota$ $y \in V \setminus V_1$.

Απόδειξη (συνέχεια)

Έχουμε λοιπόν $W_{xy} > W_{uv}$ και $[u,v] \notin$ Τ.
Αντικαθιστώντας $[x, y]$ με $[u, v]$ παίρνουμε ένα νέο δένδρο επικάλυψης $\mathrm{A}' = (\mathrm{V}, \mathrm{E}' \cup [u, v] \setminus [x, y])$ τέτοιο ώστε :

$$\text{Κόστος } (\mathrm{A}') < \text{Κόστος } (\mathrm{A})$$

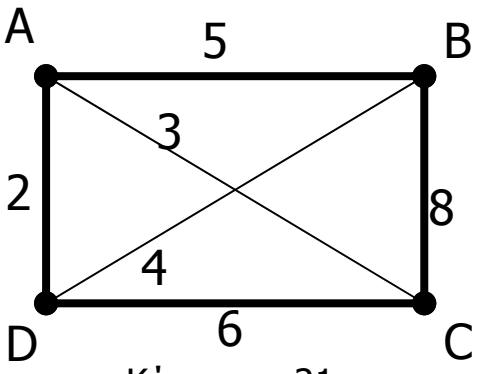
το οποίο είναι σε αντίφαση με την (*).

Το ΔΕΕΚ μπορεί να δώσει ένα κάτω φράγμα της βέλτιστης λύσης του TSP

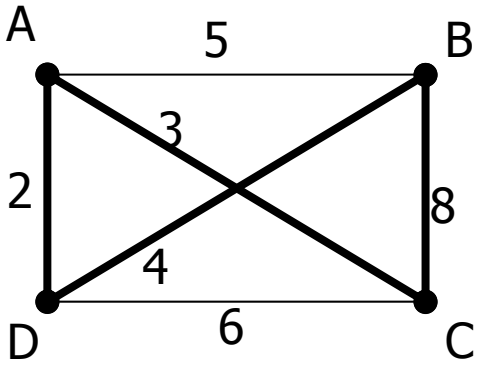
TSP:

Δίνεται ένας πλήρης γράφος G=(V, E, W) με βάρη, τάξης n. Να βρεθεί ένας κύκλος (Hamilton) ο οποίος να περνά από όλους τους κόμβους μία και μόνο μία φορά και ο οποίος να έχει ελάχιστο κόστος.

Πρόβλημα πλανόδιου πωλητή (TSP)

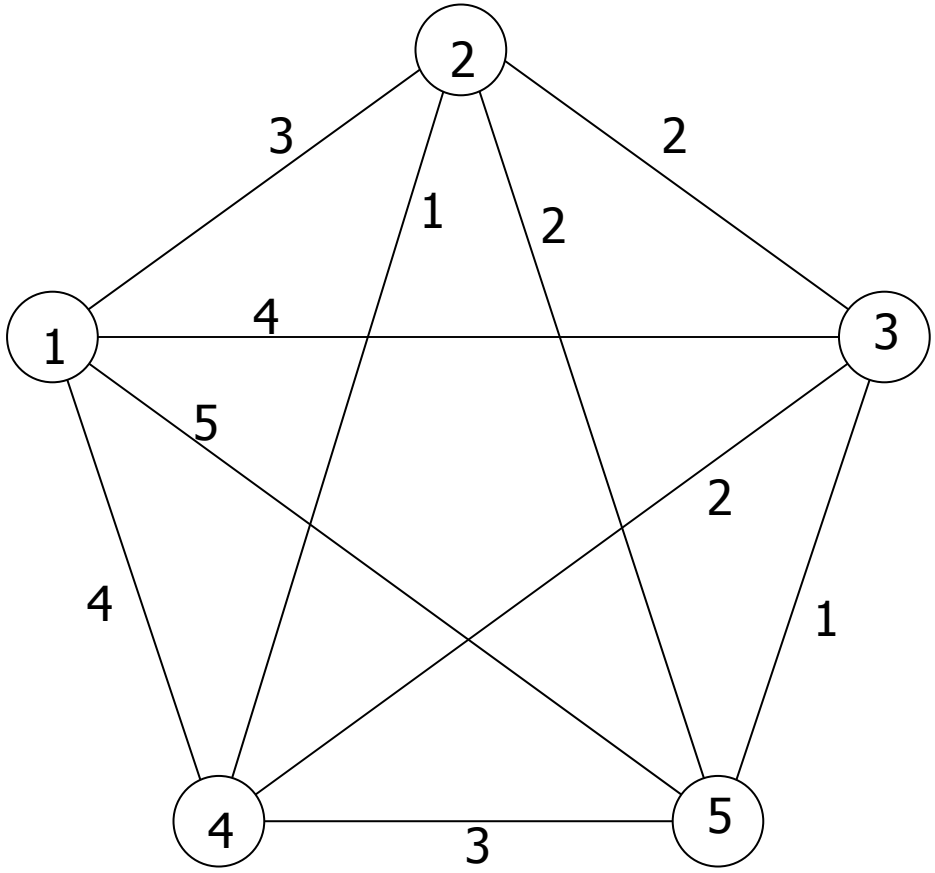


Κόστος = 21



Κόστος = 17

Πρόβλημα πλανόδιου πωλητή (TSP) - Παράδειγμα

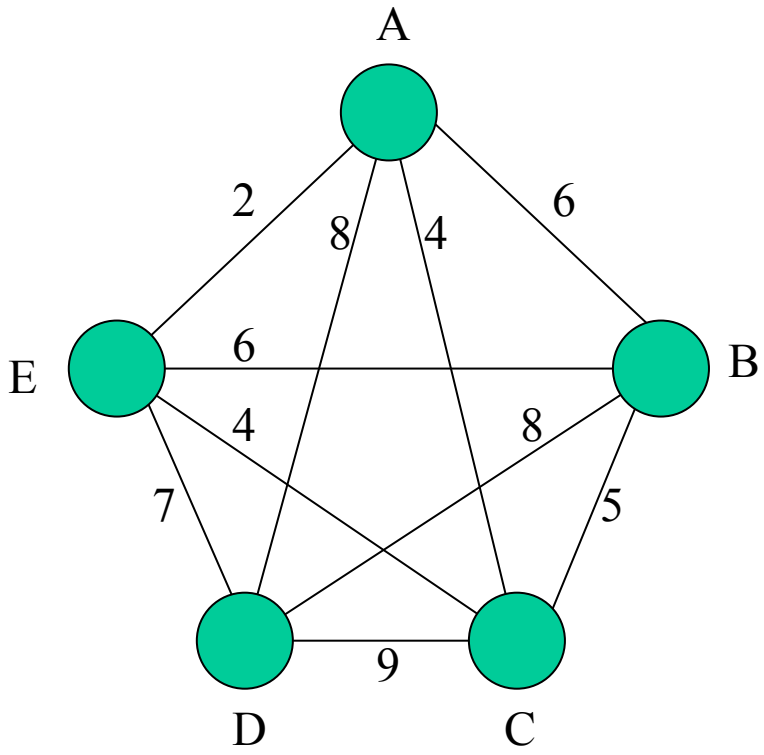


	1	**2**	**3**	**4**	**5**
1	∞	3	4	5	4
2	3	∞	2	2	1
3	4	2	∞	1	2
4	5	2	1	∞	3
5	4	1	2	3	∞

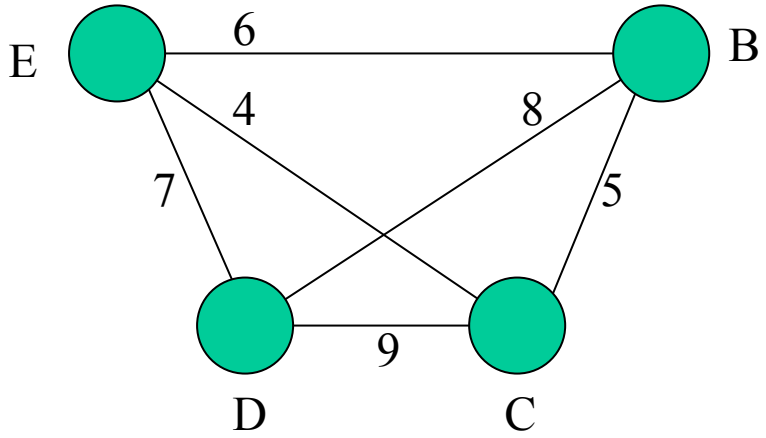
Κάτω φράγμα **B(I)**, όπου **I** ένα instance του TSP

- Αφαίρεσε ένα οποιοδήποτε κόμβο v_i από το γράφο G.
- Εύρεση ενός ΔΕΕΚ T στον υπογράφο $G' = (V\backslash\{v_i\}, E')$. Έστω K(T) το κόστος του δένδρου.
- Θεώρησε δύο πλευρές $[v_i, v_k]$ και $[v_i, v_l]$,

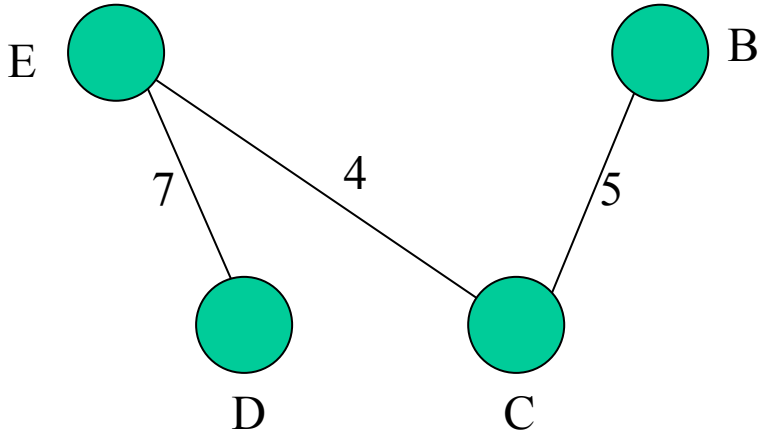
  $k, l \neq i$, με το μικρότερο κόστος
- $B(I) = K(T) + w_{il} + w_{ik}$



Αφαίρεση του A



K(T) = 16



$$B(I) = K(T) + W_{AE} + W_{AC}$$

$$= 16 + 2 + 4$$

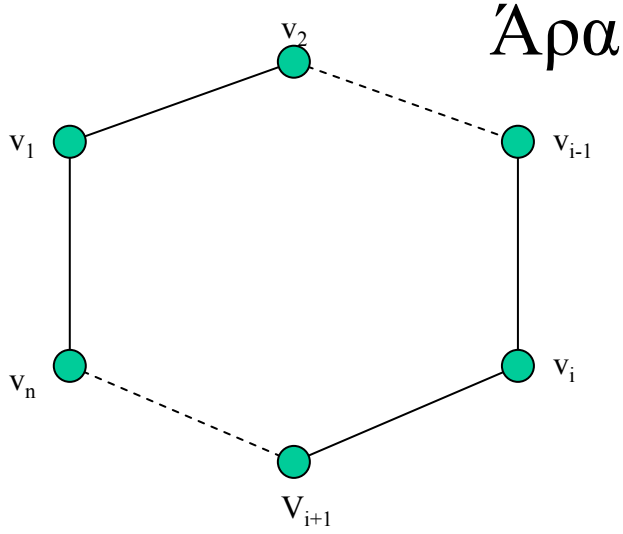
$$= 22$$

Έστω $[v_1, v_2, ..., v_i, ..., v_n, v_1]$ ένας βέλτιστος κύκλος με κόστος C_{opt}

Όταν αφαιρούμε τον κόμβο v_i, έχουμε ένα δένδρο T'

(μία αλυσίδα $v_{i-1}, \ldots, v_2, v_1, v_n, v_{n-1}, ..., v_{i+1}$)

Προφανώς Κόστος(T) ≤ Κόστος(T')



Άρα $B(I) = \text{Κόστος}(T) + W_{ik} + W_{il}$

$\leq \text{Κόστος}(T') + W_{i,i-1} + W_{i,\, i+1}$

$= \text{Κόστος}(\text{Βέλτιστου κύκλου})$

$= C_{opt}$