

# Θεμελιώδη Θέματα Επιστήμης Υπολογιστών

ΣΗΜΜΥ – ΣΕΜΦΕ ΕΜΠ

---

5<sup>η</sup> ενότητα:

Αλγόριθμοι γράφων και δικτύων

Επιμέλεια διαφανειών:

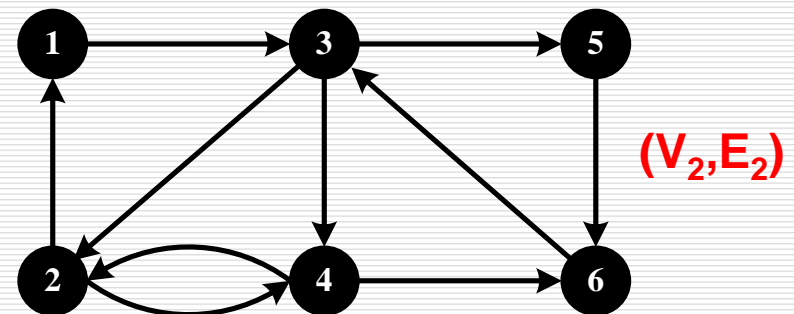
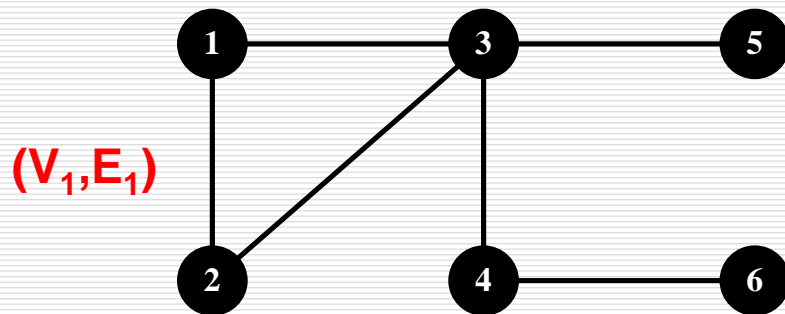
Στάθης Ζάχος, Άρης Παγουρτζής, Δημήτρης Φωτάκης

Ευχαριστίες: μέρος των διαφανειών προέρχεται από διαφάνειες του συναδέλφου Σταύρου Νικολόπουλου (Παν. Ιωαννίνων)



Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Εθνικό Μετσόβιο Πολυτεχνείο

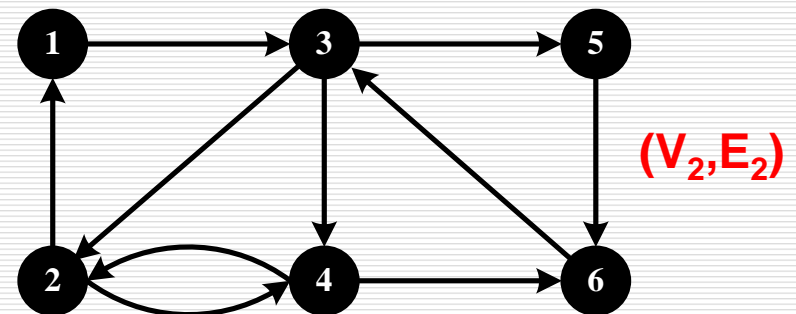
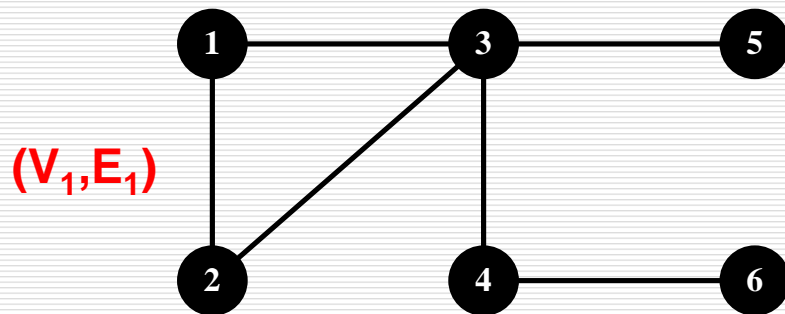
# Γράφοι (επανάληψη)



- **Γράφος (ή γράφημα):** ζεύγος  $(V, E)$ ,  $V$  ένα μη κενό σύνολο,  $E$  διμελής σχέση πάνω στο  $V$
- **Μη κατευθυνόμενος γράφος:** σχέση  $E$  συμμετρική
- **$V$ : κορυφές (vertices), κόμβοι (nodes)**
- **$E$ : ακμές (edges)**
  - $E_1 = \{\{1,2\}, \{1,3\}, \{2,3\}, \{3,4\}, \{3,5\}, \{4,6\}\}$
  - $E_2 = \{(1,3), (2,1), (2,4), (3,2), (3,4), (3,5), (4,2), (4,6), (5,6), (6,3)\}$

# Γράφοι (επανάληψη): ορολογία

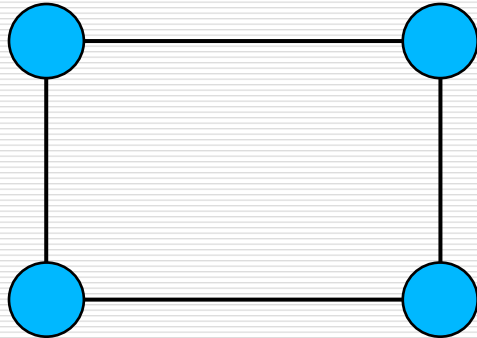
---



- **Γειτονικές (adjacent) κορυφές:** συνδέονται με ακμή, π.χ. 4 και 6
- **Άκρα (endpoints) ακμής**
- **Προσπίπτουσα (incident) ακμή (σε κόμβο)**
- **Γειτονικές ακμές**

# Γράφοι (επανάληψη): ορολογία

---

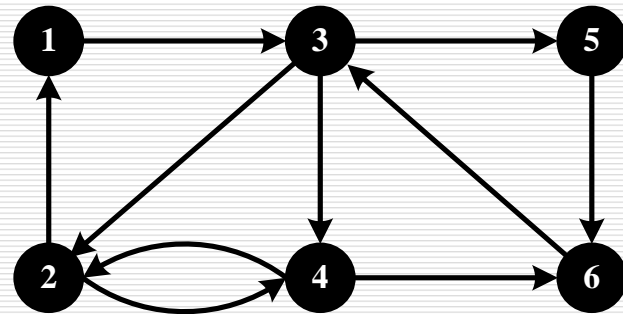
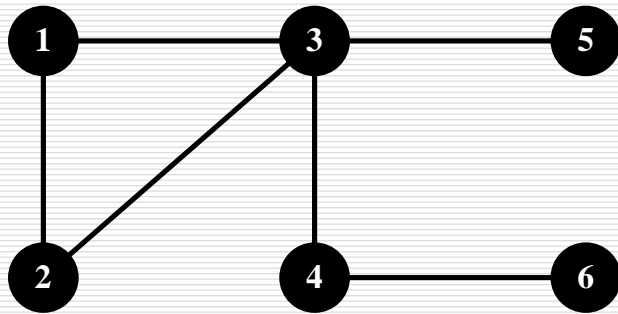


2-κανονικός γράφος

- **Βαθμός (degree, valence) κορυφής  $v$ :** ο αριθμός των ακμών που προσπίπτουν στην  $v$ ,  $\text{deg}(v)$
- Ένας (μη κατευθυνόμενος) γράφος όπου  $\text{deg}(v)=k$  για κάθε κορυφή  $v$ , λέγεται  **$k$ -κανονικός ( $k$ -regular)**
- Σημαντική ιδιότητα:  $\sum \text{deg}(v) = 2|E|$
- Σε κατευθυνόμενο γράφο:  **$\text{in-deg}(v)$ ,  $\text{out-deg}(v)$**

# Γράφοι (επανάληψη): διαδρομές

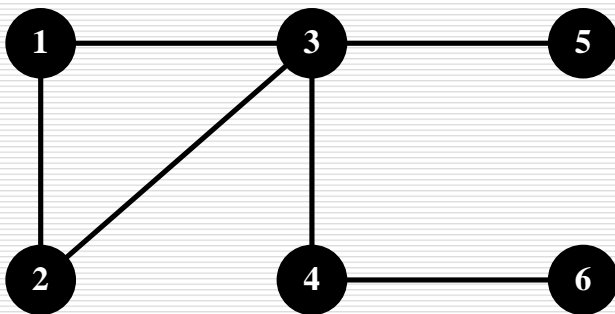
---



- **Δρόμος:** έγκυρη ακολουθία από κορυφές-ακμές
- **Μονοπάτι:** δρόμος χωρίς επαναλήψεις ακμών
  - **Απλό μονοπάτι:** μονοπάτι χωρίς επαναλήψεις κορυφών
- **Κύκλος:** κλειστό μονοπάτι
  - **Απλός κύκλος:** απλό κλειστό μονοπάτι
- **Μήκος δρόμου:** το πλήθος των ακμών του

# Γράφοι (επανάληψη): αναπαράσταση

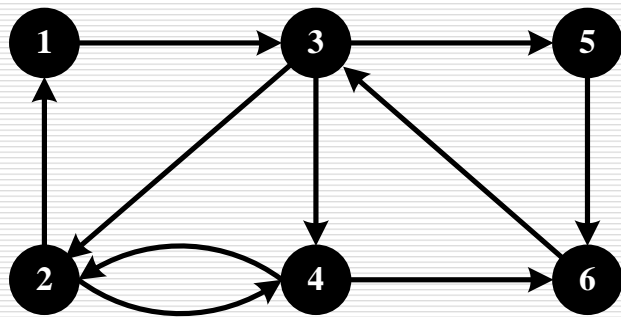
- ... με **πίνακα γειτνίασης**:  $A[i, j] = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$ 
  - Αν έχουμε βάρη,  $A[i, j] = w(v_i, v_j)$
  - Μη-κατευθυνόμενος: **συμμετρικός** πίνακας
  - Χώρος:  $\Theta(n^2)$
  - Προσπέλαση γειτόνων:  $\Theta(n)$
  - Άμεσος έλεγχος ύπαρξης ακμής:  $O(1)$



	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	1	0	0	0
3	1	1	0	1	1	0
4	0	0	1	0	0	1
5	0	0	1	0	0	0
6	0	0	0	1	0	0

# Γράφοι (επανάληψη): αναπαράσταση

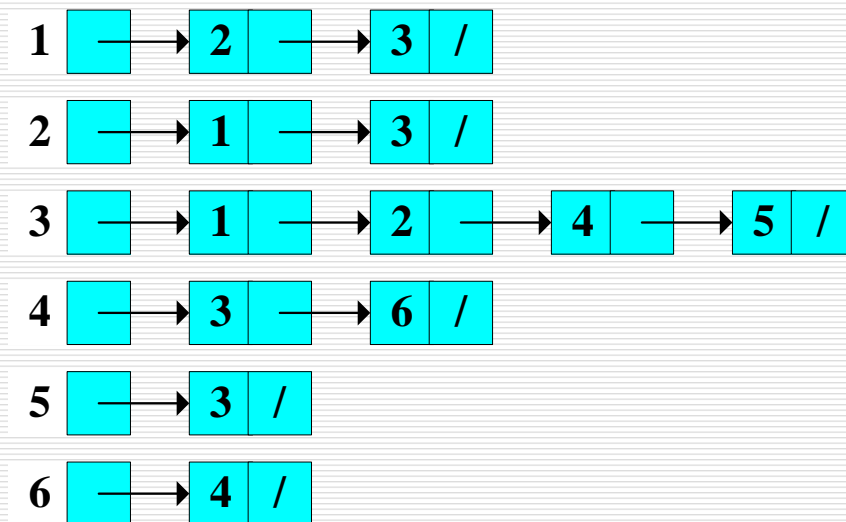
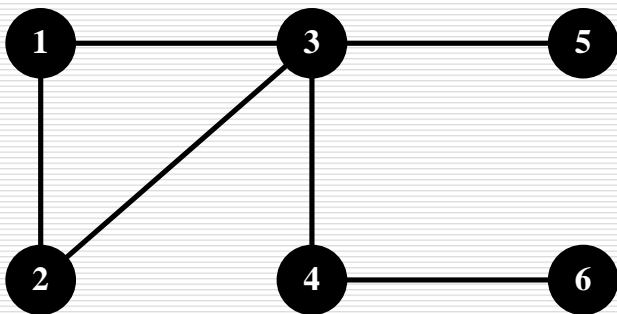
- ... με **πίνακα γειτνίασης**:  $A[i, j] = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$ 
  - Αν έχουμε βάρη,  $A[i, j] = w(v_i, v_j)$
  - Κατευθυνόμενος: **μη-συμμετρικός** πίνακας
  - Χώρος:  $\Theta(n^2)$
  - Προσπέλαση γειτόνων:  $\Theta(n)$
  - Άμεσος έλεγχος ύπαρξης ακμής:  $O(1)$



	1	2	3	4	5	6
1	0	0	1	0	0	0
2	1	0	0	1	0	0
3	0	1	0	1	1	0
4	0	1	0	0	0	1
5	0	0	0	0	0	1
6	0	0	1	0	0	0

# Γράφοι (επανάληψη): αναπαράσταση

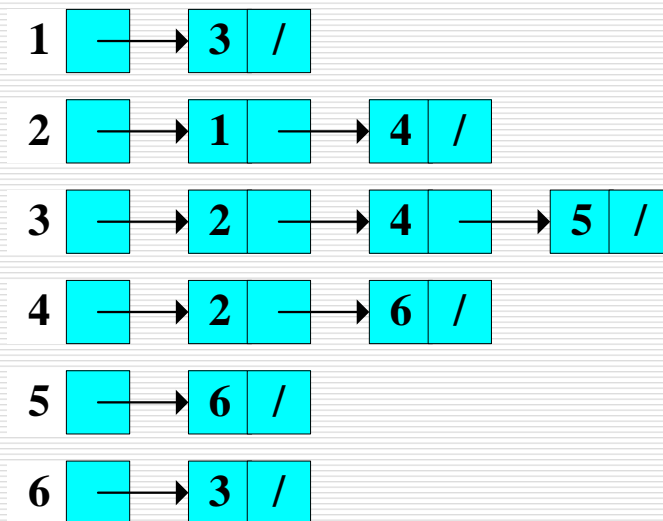
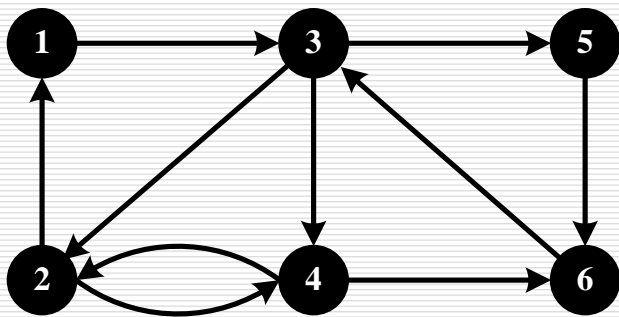
- ... με **λίστες γειτνίασης**: γειτονικές κορυφές σε λίστες
  - Αν έχουμε βάρη, τα αποθηκεύουμε στους κόμβους
  - Χώρος:  $\Theta(m)$
  - Προσπέλαση γειτόνων:  $\Theta(\text{deg}(u))$
  - Έλεγχος ύπαρξης ακμής:  $O(\text{deg}(u))$





# Γράφοι (επανάληψη): αναπαράσταση

- ... με **λίστες γειτνίασης**: γειτονικές κορυφές σε λίστες
  - Αν έχουμε βάρη, τα αποθηκεύουμε στους κόμβους
  - Χώρος:  $\Theta(m)$
  - Προσπέλαση γειτόνων:  $\Theta(\text{deg}(u))$
  - Έλεγχος ύπαρξης ακμής:  $O(\text{deg}(u))$



# Γράφοι (επανάληψη): συνεκτικότητα

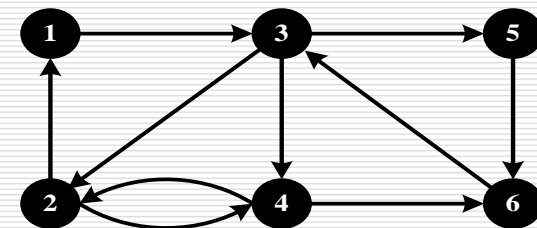
- Ένας μη κατευθυνόμενος γράφος λέγεται **συνεκτικός (connected)** αν υπάρχει δρόμος μεταξύ οποιωνδήποτε δύο κορυφών του

Σε συνεκτικό γράφο ισχύει:  $n - 1 \leq e \leq \frac{n(n-1)}{2}, n = |V|, e = |E|.$

- Ένας κατευθυνόμενος γράφος λέγεται

- **ισχυρά συνεκτικός (strongly connected)**

αν υπάρχει δρόμος μεταξύ οποιωνδήποτε δύο κορυφών του ακολουθώντας τις κατευθύνσεις των ακμών



- **ασθενώς συνεκτικός (weakly connected)**

αν υπάρχει δρόμος μεταξύ οποιωνδήποτε δύο κορυφών του αγνοώντας τις κατευθύνσεις των ακμών

# Συντομότερες διαδρομές (Dijkstra)

```
procedure Dijkstra;
begin (* Αρχικοποίηση *)
  S := {1}; for i:=2 to n do begin D[i]:=cost[1,i]; P[i]:=1 end;
  for i:=2 to n-1 do
  begin
    select w from V - S such that D[w] is minimum;
    S := S + {w};
    for all v in V - S do
      if D[v] > D[w] + C[w,v] then
        P[v] := w;
        D[v] := D[w]+C[w,v]
      end
    end
  end
end
```

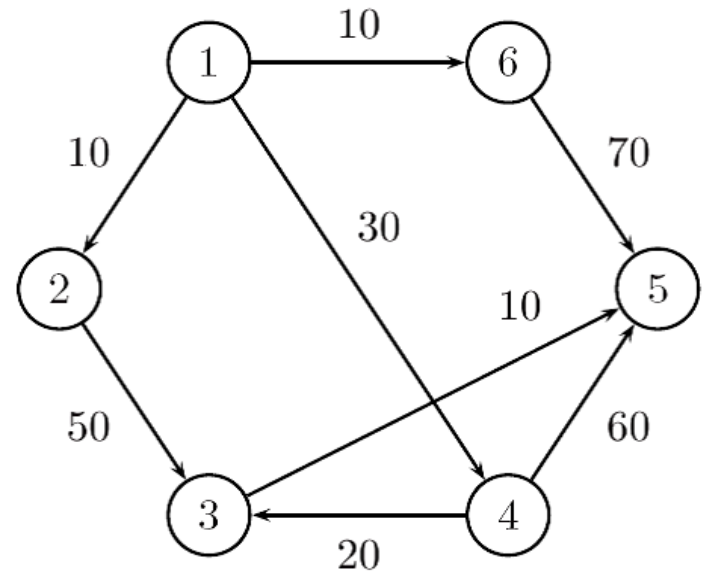
**Πολυπλ/τα**  
 **$O(|V|^2)$ :**

σε κάθε  
επανάληψη  
 **$O(|V|)$**  για  
εύρεση  
ελαχίστου,  
 **$O(|V|)$**  για  
ενημέρωση  
αποστάσεων

# Παράδειγμα Dijkstra

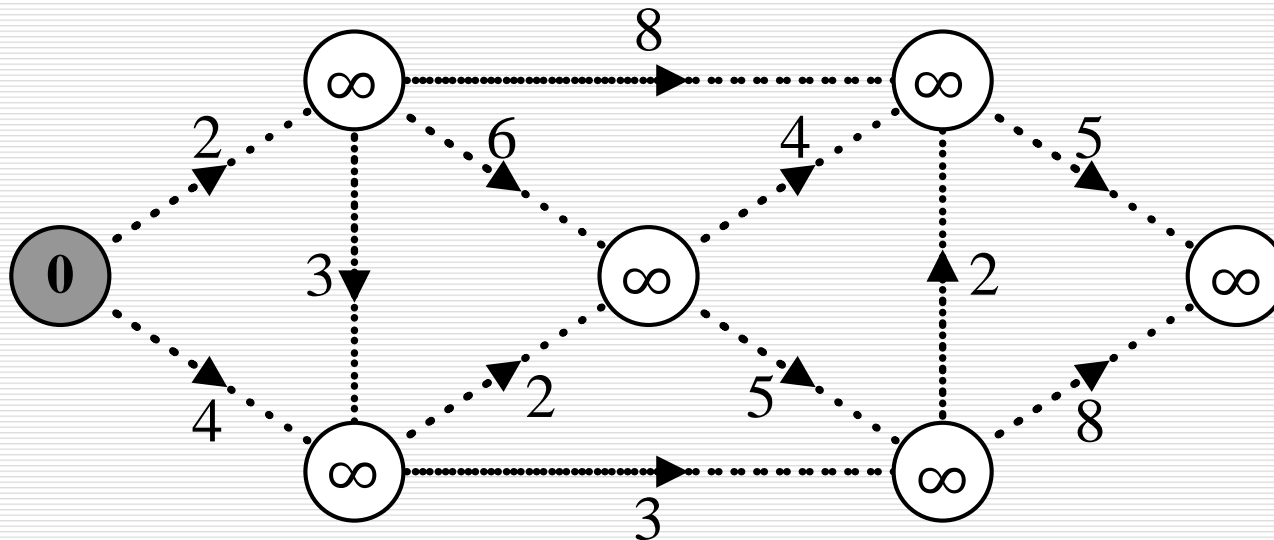
```

procedure Dijkstra;
begin (* Αρχικοποίηση *)
  S := {1}; for i:=2 to n do begin D[i]:=cost[1,i]; P[i]:=1 end;
  for i:=2 to n-1 do
  begin
    select w from V - S such that D[w] is minimum;
    S := S + {w};
    for all v in V - S do
      if D[v] > D[w] + C[w,v] then
        P[v] := w;
        D[v] := D[w]+C[w,v]
      end
    end
  end
end
  
```



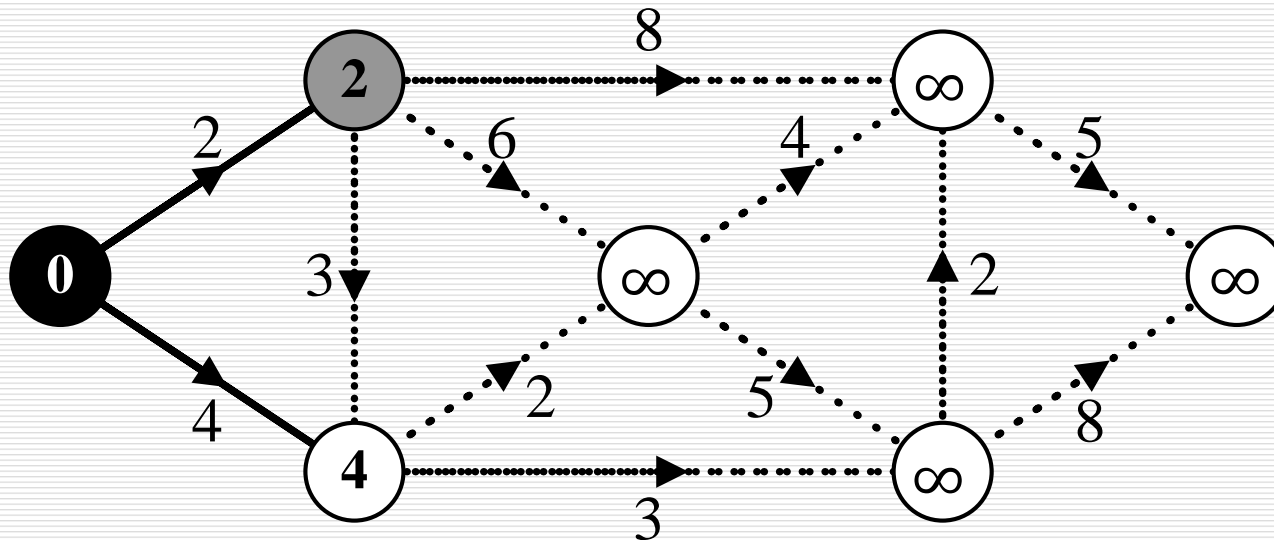
Βήμα	S	w	D					P				
			2	3	4	5	6	2	3	4	5	6
-	{1}	-	10	∞	30	∞	10	1	1	1	1	1
2	{1,2}	2		60	30	∞	10		2			
3	{1,2,6}	6		60	30	80					6	
4	{1,2,6,4}	4		50		80			4			
5	{1,2,6,4,3}	3				60					3	
6	{1,2,6,4,3,5}	5										

# 2ο παράδειγμα Dijkstra



Οι ετικέτες των κόμβων δείχνουν την μέχρι στιγμής ελάχιστη απόσταση από τον αρχικό κόμβο (πίνακας D).

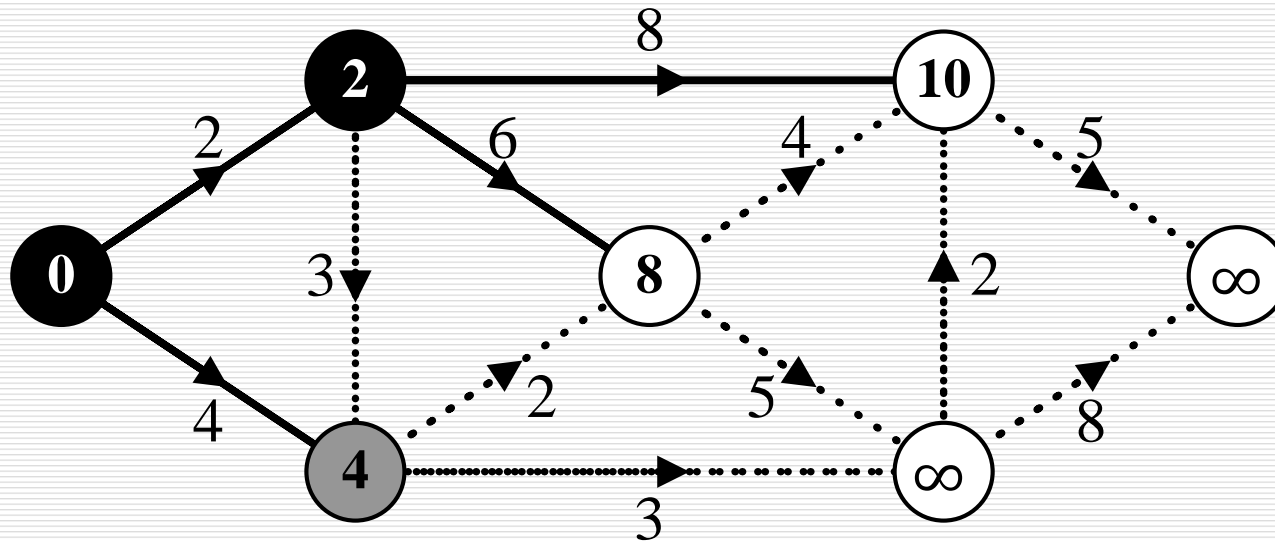
# 2ο παράδειγμα Dijkstra



Οι ετικέτες των κόμβων δείχνουν την μέχρι στιγμής ελάχιστη απόσταση από τον αρχικό κόμβο (πίνακας D), οι συνεχείς ακμές δείχνουν ποιος είναι ο αντίστοιχος προηγούμενος κόμβος (πίνακας P).

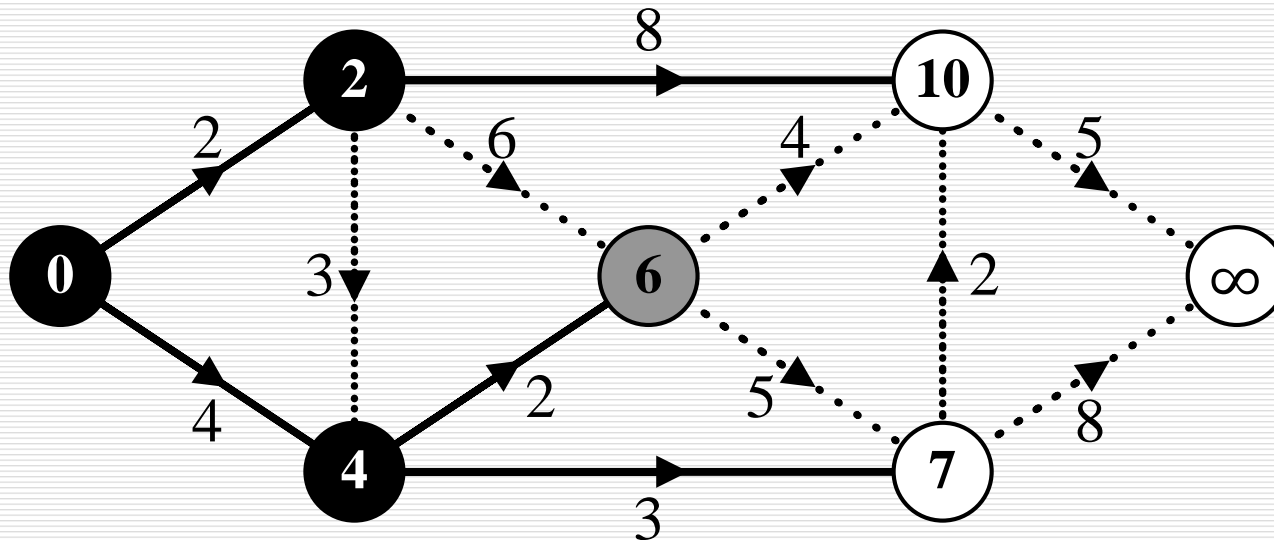
# 2ο παράδειγμα Dijkstra

---



Οι ετικέτες των κόμβων δείχνουν την μέχρι στιγμής ελάχιστη απόσταση από τον αρχικό κόμβο (πίνακας D), οι συνεχείς ακμές δείχνουν ποιος είναι ο αντίστοιχος προηγούμενος κόμβος (πίνακας P).

# 2ο παράδειγμα Dijkstra

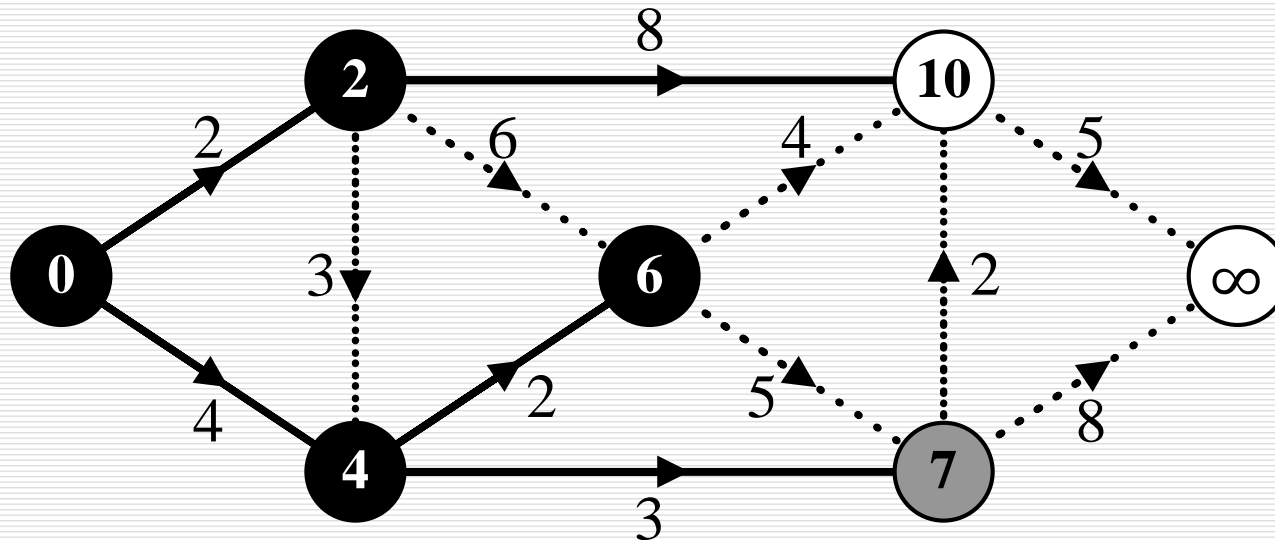


Οι ετικέτες των κόμβων δείχνουν την μέχρι στιγμής ελάχιστη απόσταση από τον αρχικό κόμβο (πίνακας D), οι συνεχείς ακμές δείχνουν ποιος είναι ο αντίστοιχος προηγούμενος κόμβος (πίνακας P).



# 2ο παράδειγμα Dijkstra

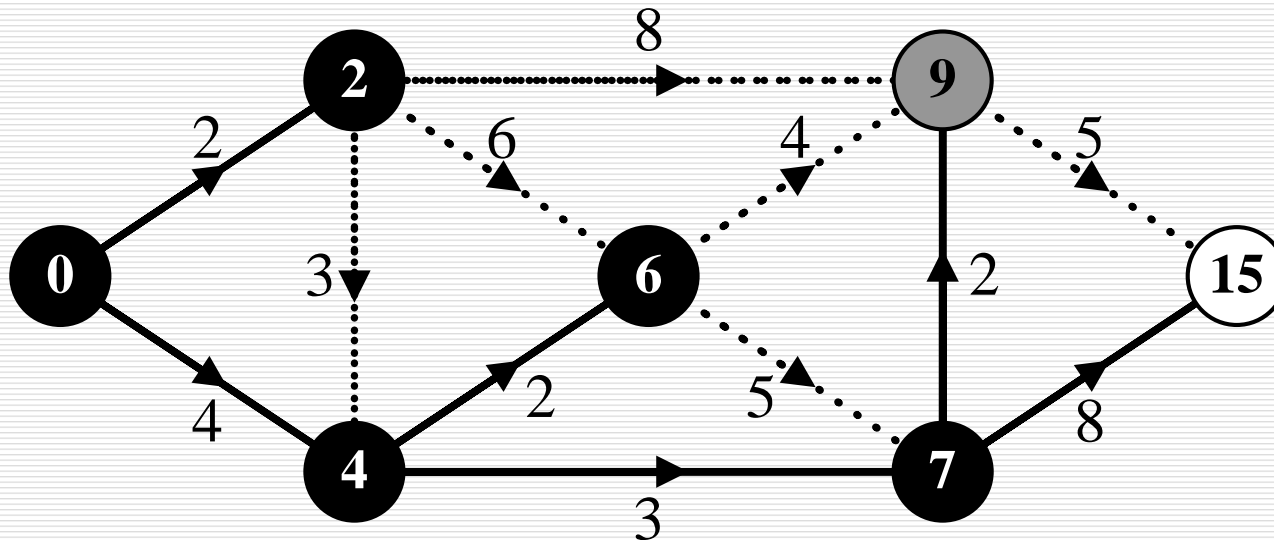
---



Οι ετικέτες των κόμβων δείχνουν την μέχρι στιγμής ελάχιστη απόσταση από τον αρχικό κόμβο (πίνακας D), οι συνεχείς ακμές δείχνουν ποιος είναι ο αντίστοιχος προηγούμενος κόμβος (πίνακας P).

# 2ο παράδειγμα Dijkstra

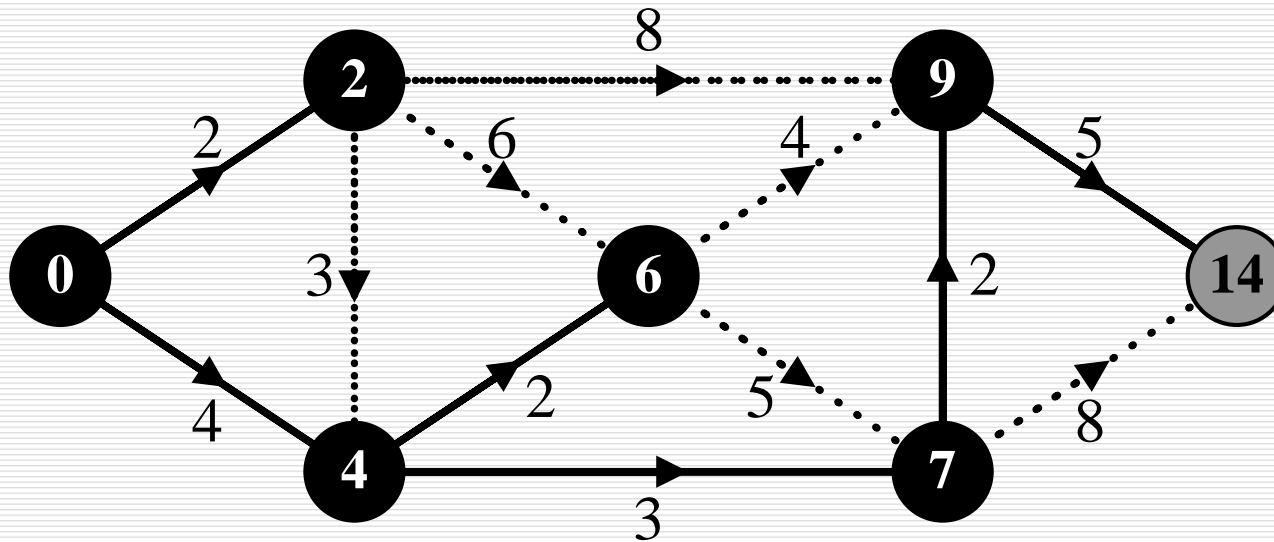
---



Οι ετικέτες των κόμβων δείχνουν την μέχρι στιγμής ελάχιστη απόσταση από τον αρχικό κόμβο (πίνακας D), οι συνεχείς ακμές δείχνουν ποιος είναι ο αντίστοιχος προηγούμενος κόμβος (πίνακας P).

# 2ο παράδειγμα Dijkstra

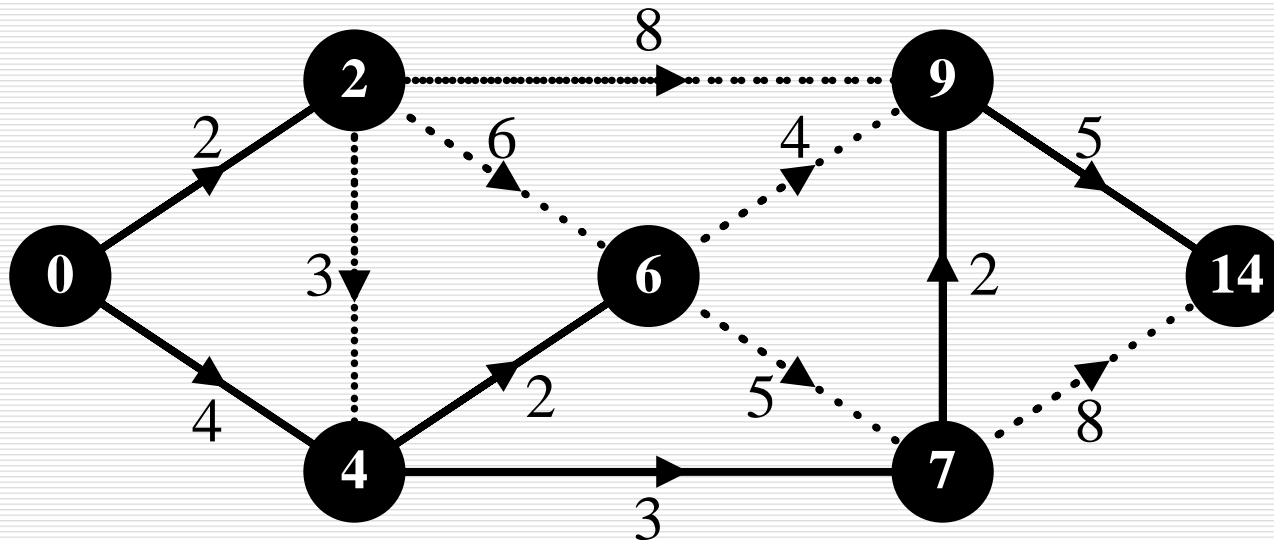
---



Οι ετικέτες των κόμβων δείχνουν την μέχρι στιγμής ελάχιστη απόσταση από τον αρχικό κόμβο (πίνακας D), οι συνεχείς ακμές δείχνουν ποιος είναι ο αντίστοιχος προηγούμενος κόμβος (πίνακας P).

# 2ο παράδειγμα Dijkstra

---



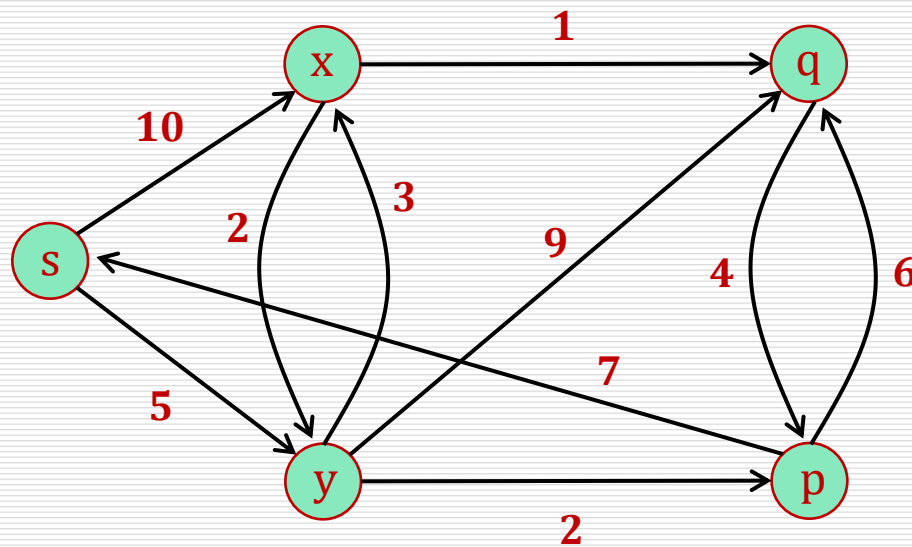
Οι ετικέτες των κόμβων δείχνουν την μέχρι στιγμής ελάχιστη απόσταση από τον αρχικό κόμβο (πίνακας D), οι συνεχείς ακμές δείχνουν ποιος είναι ο αντίστοιχος προηγούμενος κόμβος (πίνακας P).

# 3ο παράδειγμα Dijkstra



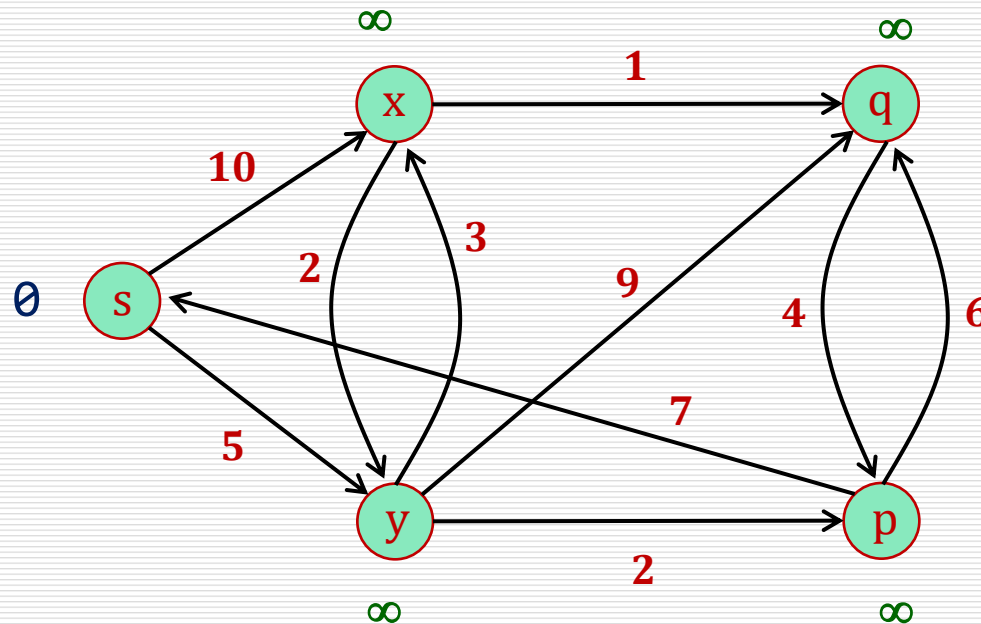
Παράδειγμα

Είσοδος Αλγόριθμου



# 3ο παράδειγμα Dijkstra

## Παράδειγμα



Initialize(G,s)

για κάθε κόμβο  $v \in V$ :

$d(v) \leftarrow \infty$

$prev(v) \leftarrow NIL$

$d(s) \leftarrow 0$

$d(s)=0$

$d(x)=\infty$

$d(y)=\infty$

$d(p)=\infty$

$d(q)=\infty$

$prev(s)=NIL$

$prev(x)=NIL$

$prev(y)=NIL$

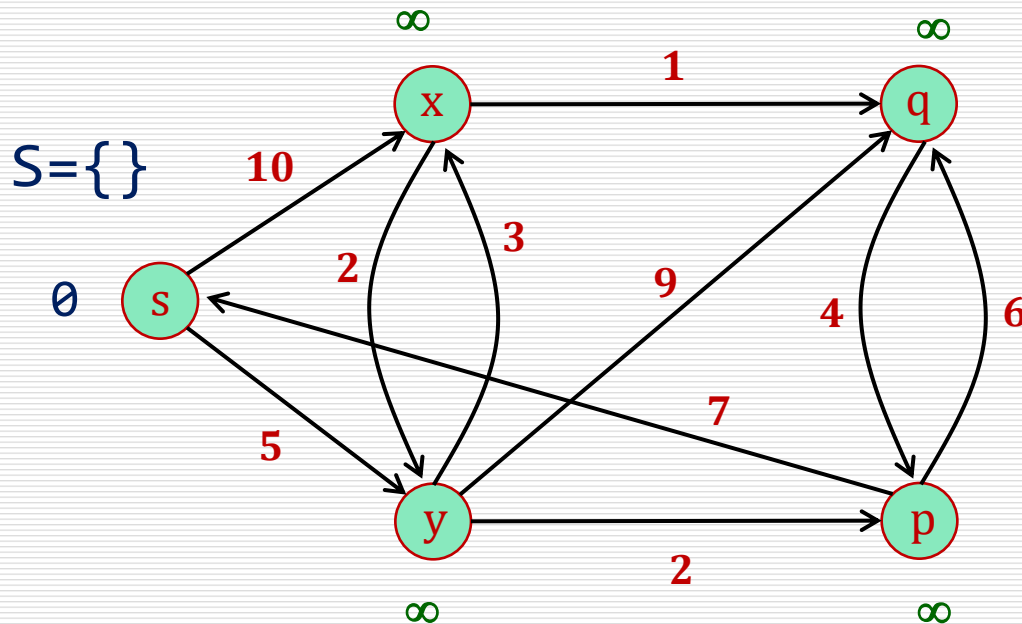
$prev(p)=NIL$

$prev(q)=NIL$

# 3ο παράδειγμα Dijkstra

▣ Παράδειγμα

$Q = \{s, x, y, p, q\}$



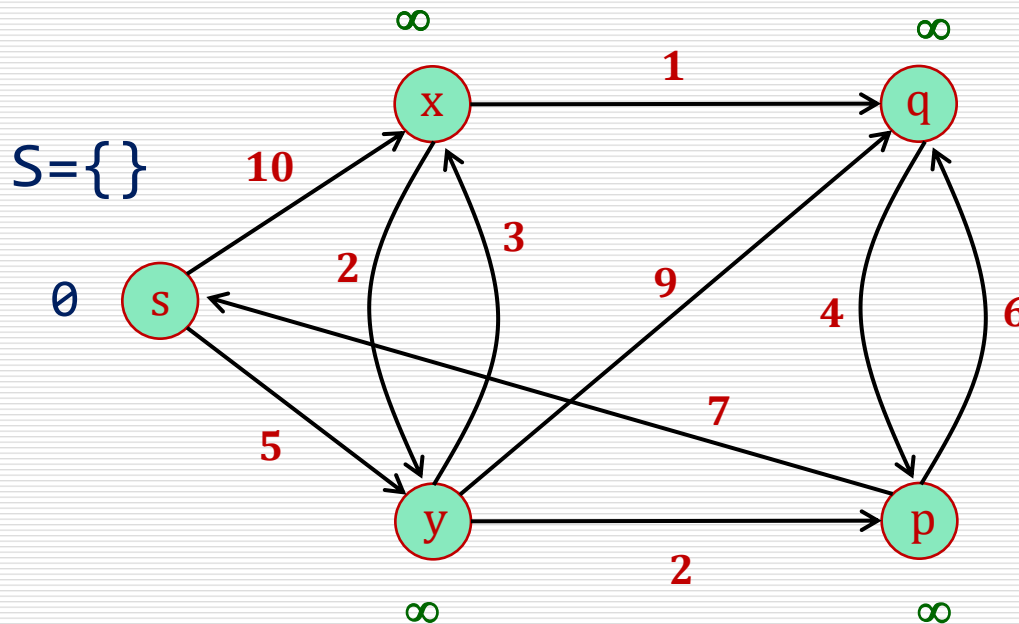
---

$d(s)=0$	$d(x)=\infty$	$d(y)=\infty$	$d(p)=\infty$	$d(q)=\infty$
$prev(s)=NIL$	$prev(x)=NIL$	$prev(y)=NIL$	$prev(p)=NIL$	$prev(q)=NIL$

# 3ο παράδειγμα Dijkstra

■ Παράδειγμα

$Q = \{s, x, y, p, q\}$   
while  $Q \neq \emptyset$ :



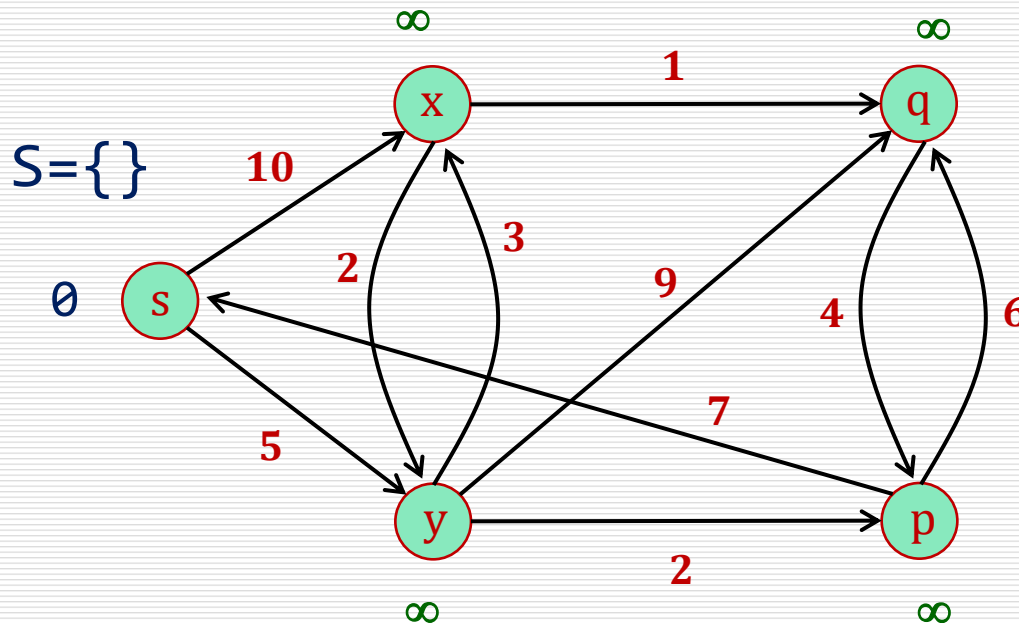
---

$d(s)=0$	$d(x)=\infty$	$d(y)=\infty$	$d(p)=\infty$	$d(q)=\infty$
$prev(s)=NIL$	$prev(x)=NIL$	$prev(y)=NIL$	$prev(p)=NIL$	$prev(q)=NIL$



# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{s, x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

---

$d(s) = 0$	$d(x) = \infty$	$d(y) = \infty$	$d(p) = \infty$	$d(q) = \infty$
$\text{prev}(s) = \text{NIL}$	$\text{prev}(x) = \text{NIL}$	$\text{prev}(y) = \text{NIL}$	$\text{prev}(p) = \text{NIL}$	$\text{prev}(q) = \text{NIL}$

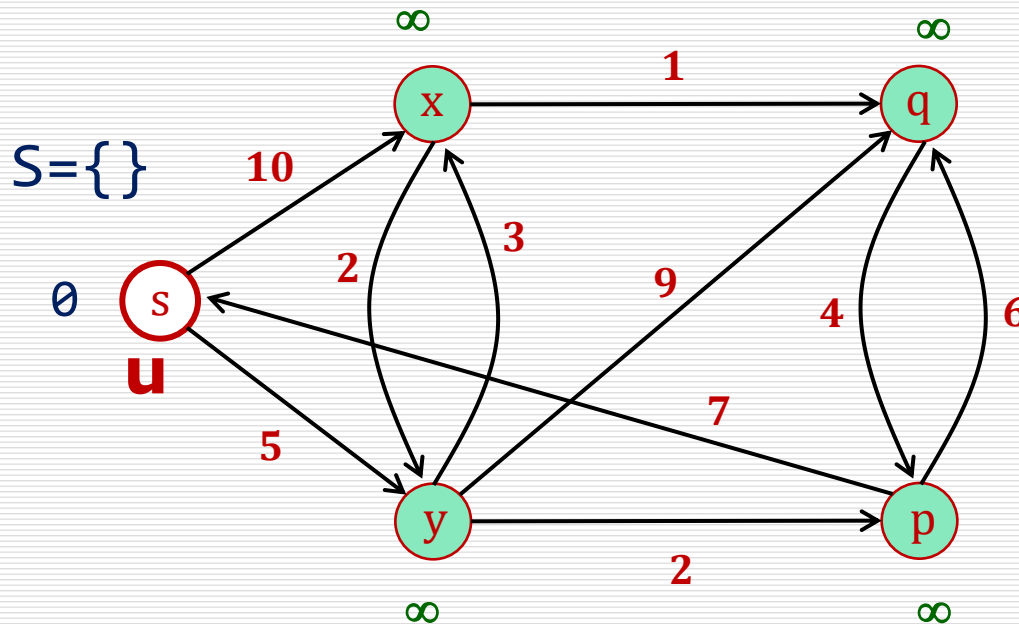
# 3ο παράδειγμα Dijkstra

## Παράδειγμα

$Q = \{x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$



$d(s) = 0$

$d(x) = \infty$

$d(y) = \infty$

$d(p) = \infty$

$d(q) = \infty$

$\text{prev}(s) = \text{NIL}$

$\text{prev}(x) = \text{NIL}$

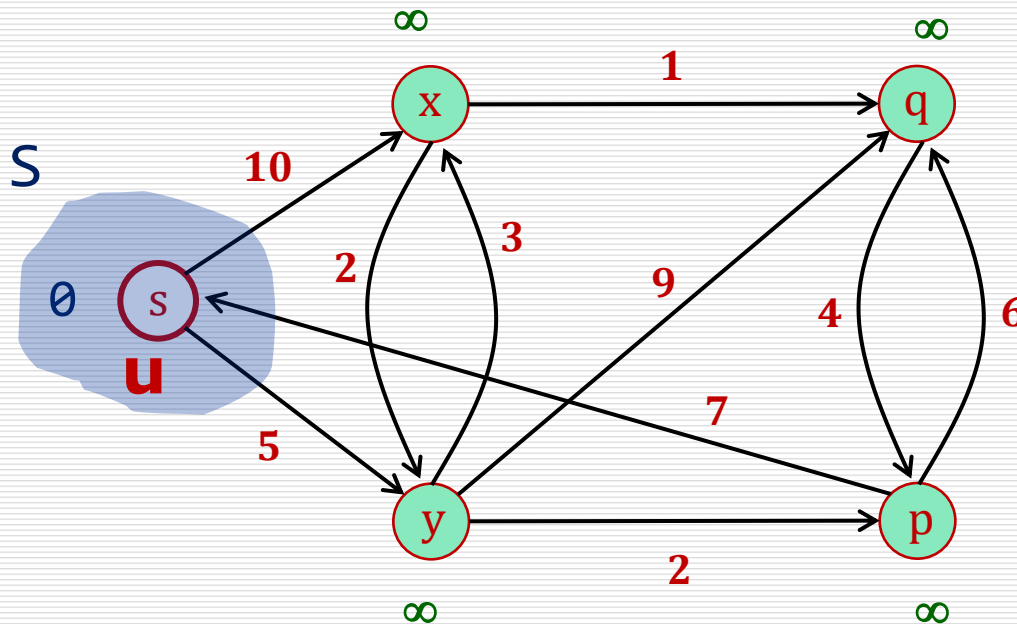
$\text{prev}(y) = \text{NIL}$

$\text{prev}(p) = \text{NIL}$

$\text{prev}(q) = \text{NIL}$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

$S \leftarrow S \cup \{u\}$

$d(s)=0$

$d(x)=\infty$

$d(y)=\infty$

$d(p)=\infty$

$d(q)=\infty$

$\text{prev}(s)=\text{NIL}$

$\text{prev}(x)=\text{NIL}$

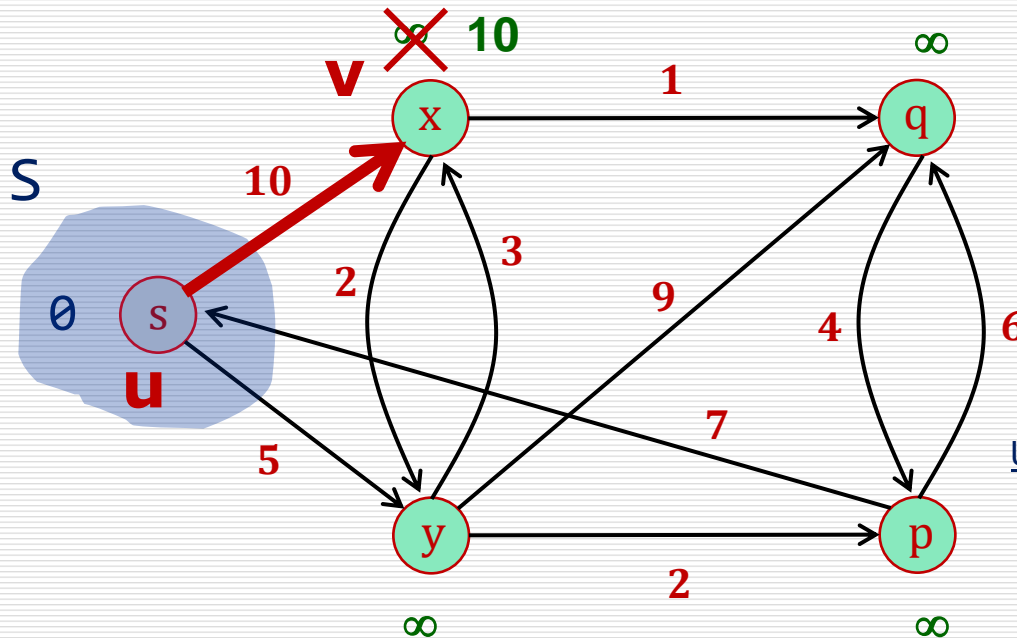
$\text{prev}(y)=\text{NIL}$

$\text{prev}(p)=\text{NIL}$

$\text{prev}(q)=\text{NIL}$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

$S \leftarrow S \cup \{u\}$

για κάθε κόμβο  $v \in N(u)$

$\text{Update}(u, v, c)$

$\text{Update}(u, v, c)$

if  $d(v) > d(u) + c(u, v)$  then

$d(v) \leftarrow d(u) + c(u, v)$

$\text{prev}(v) \leftarrow u$

$d(s) = 0$

$d(x) = 10$

$d(y) = \infty$

$d(p) = \infty$

$d(q) = \infty$

$\text{prev}(s) = \text{NIL}$

$\text{prev}(x) = s$

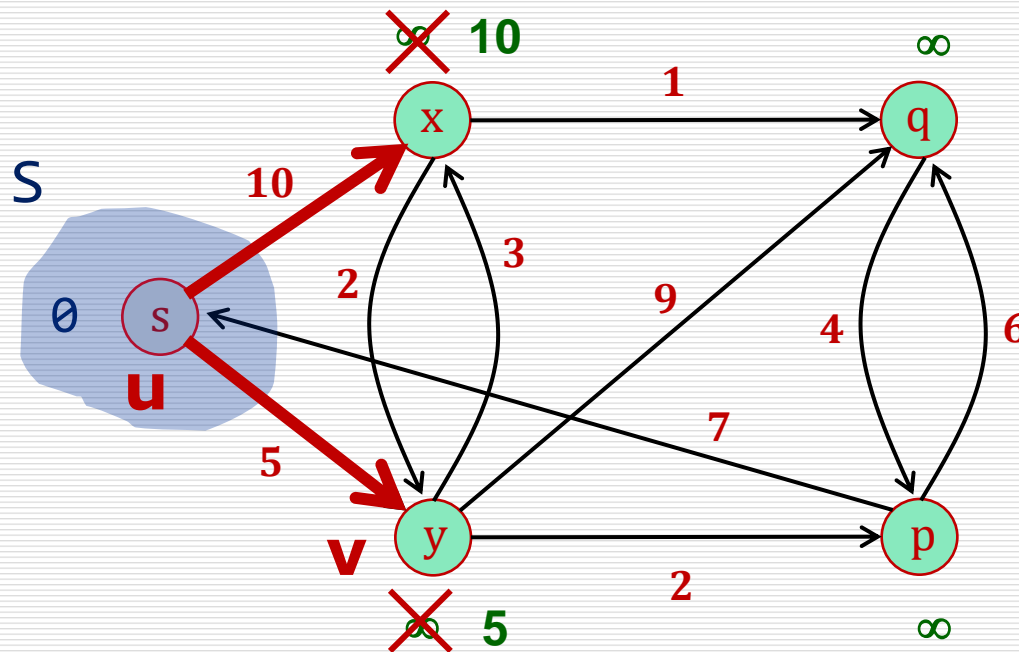
$\text{prev}(y) = \text{NIL}$

$\text{prev}(p) = \text{NIL}$

$\text{prev}(q) = \text{NIL}$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;

$S \leftarrow S \cup \{u\}$

για κάθε κόμβο  $v \in N(u)$

Update( $u, v, c$ )

$d(s) = 0$

$d(x) = 10$

$d(y) = 5$

$d(p) = \infty$

$d(q) = \infty$

prev( $s$ ) = NIL

prev( $x$ ) =  $s$

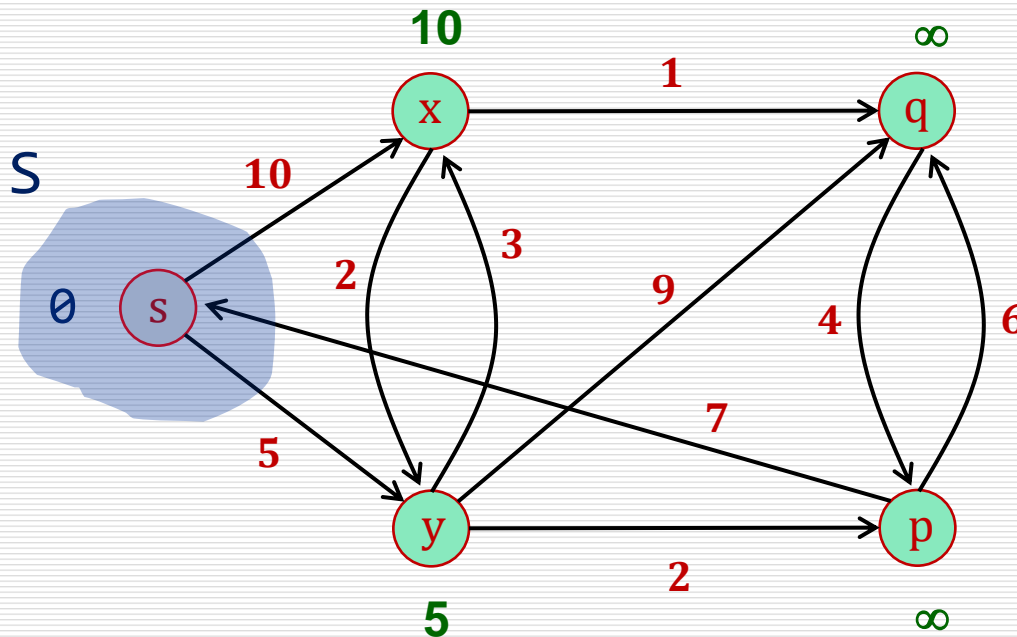
prev( $y$ ) =  $s$

prev( $p$ ) = NIL

prev( $q$ ) = NIL

# 3ο παράδειγμα Dijkstra

■ Παράδειγμα



$Q = \{x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

---

$d(s)=0$	$d(x)=10$	$d(y)=5$	$d(p)=\infty$	$d(q)=\infty$
$\text{prev}(s)=\text{NIL}$	$\text{prev}(x)=s$	$\text{prev}(y)=s$	$\text{prev}(p)=\text{NIL}$	$\text{prev}(q)=\text{NIL}$

# 3ο παράδειγμα Dijkstra

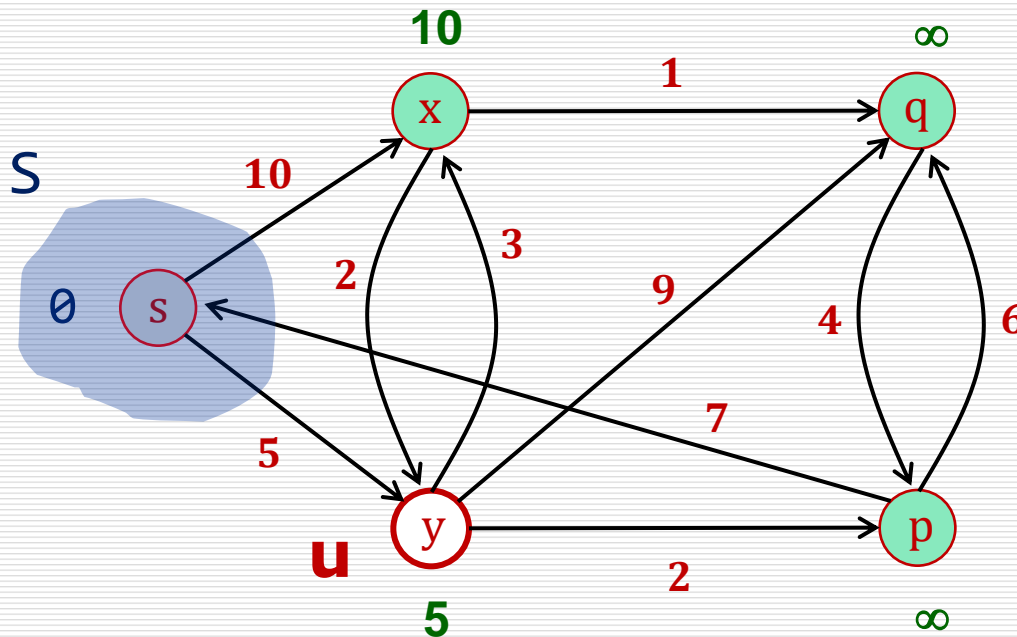
$d(s)=0$        $d(y)=5$   
 ~~$prev(s)=NIL$        $prev(y)=s$~~

▣ Παράδειγμα

$Q = \{x, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

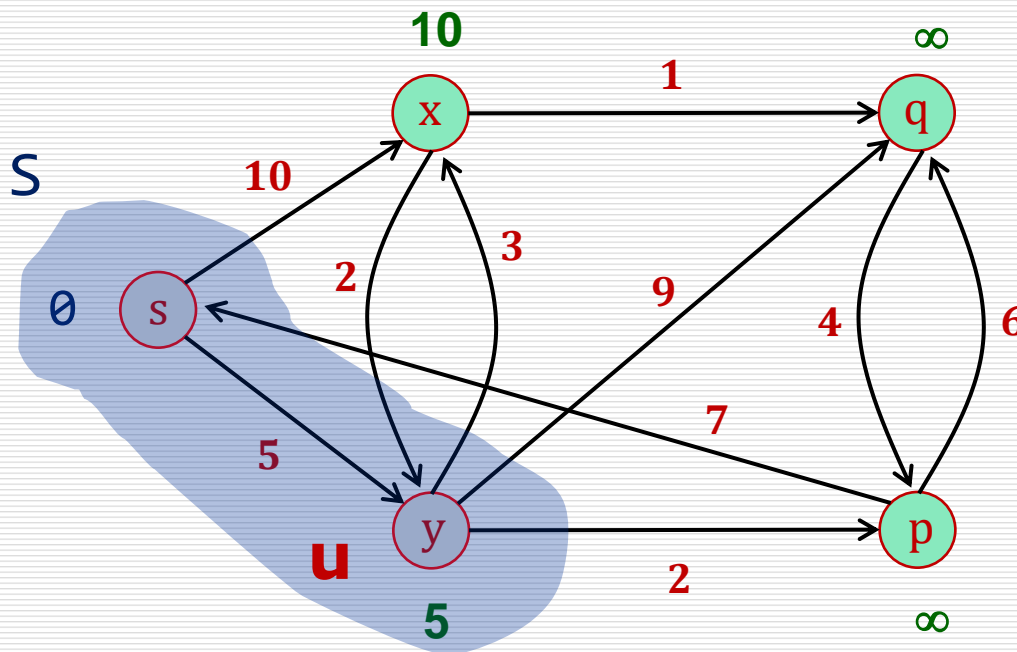


---

$d(s)=0$	$d(x)=10$	$d(y)=5$	$d(p)=\infty$	$d(q)=\infty$
$prev(s)=NIL$	$prev(x)=s$	$prev(y)=s$	$prev(p)=NIL$	$prev(q)=NIL$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;

$S \leftarrow S \cup \{u\}$

$d(s)=0$

$d(x)=10$

$d(y)=5$

$d(p)=\infty$

$d(q)=\infty$

$\text{prev}(s)=\text{NIL}$

$\text{prev}(x)=s$

$\text{prev}(y)=s$

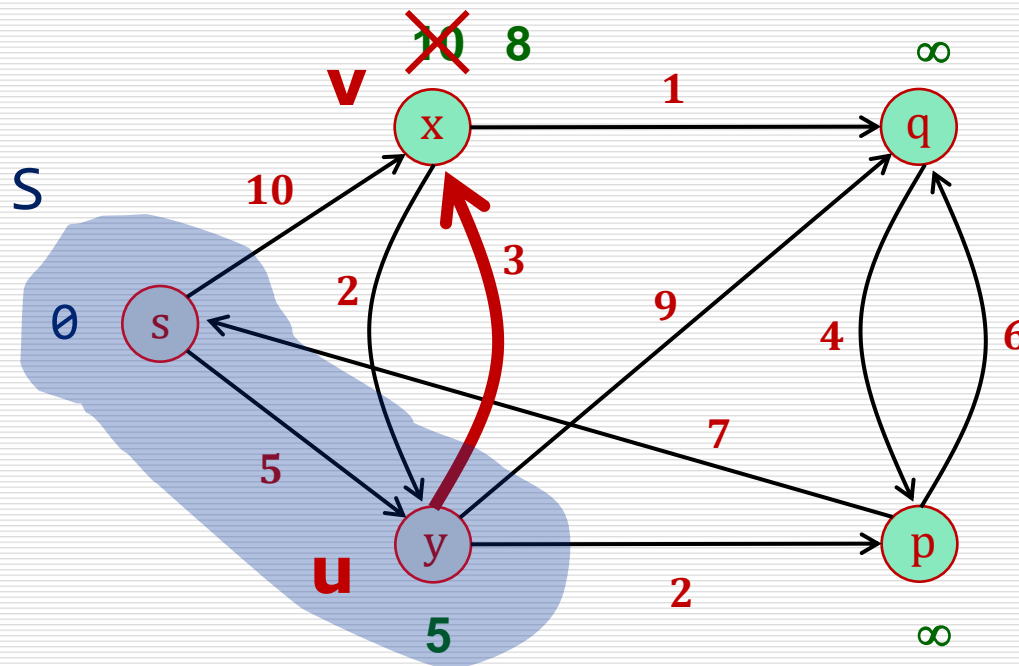
$\text{prev}(p)=\text{NIL}$

$\text{prev}(q)=\text{NIL}$



# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

$S \leftarrow S \cup \{u\}$

για κάθε κόμβο  $v \in N(u)$

Update( $u, v, c$ )

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=\infty$

$d(q)=\infty$

$prev(s)=NIL$

$prev(x)=y$

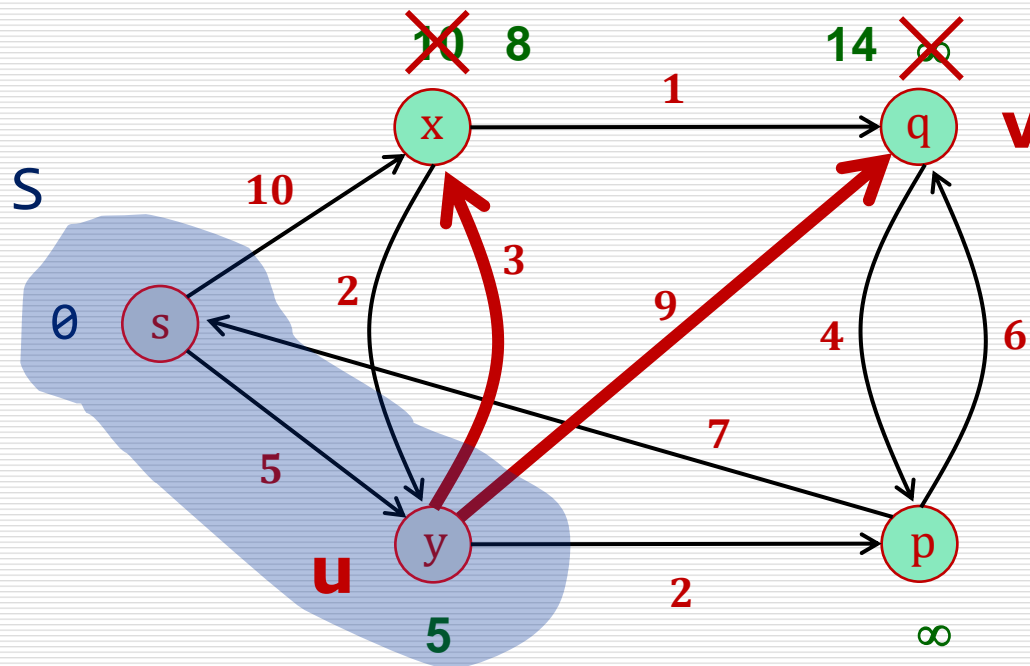
$prev(y)=s$

$prev(p)=NIL$

$prev(q)=NIL$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

$S \leftarrow S \cup \{u\}$

για κάθε κόμβο  $v \in N(u)$

Update( $u, v, c$ )

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=\infty$

$d(q)=14$

prev(s)=NIL

prev(x)=y

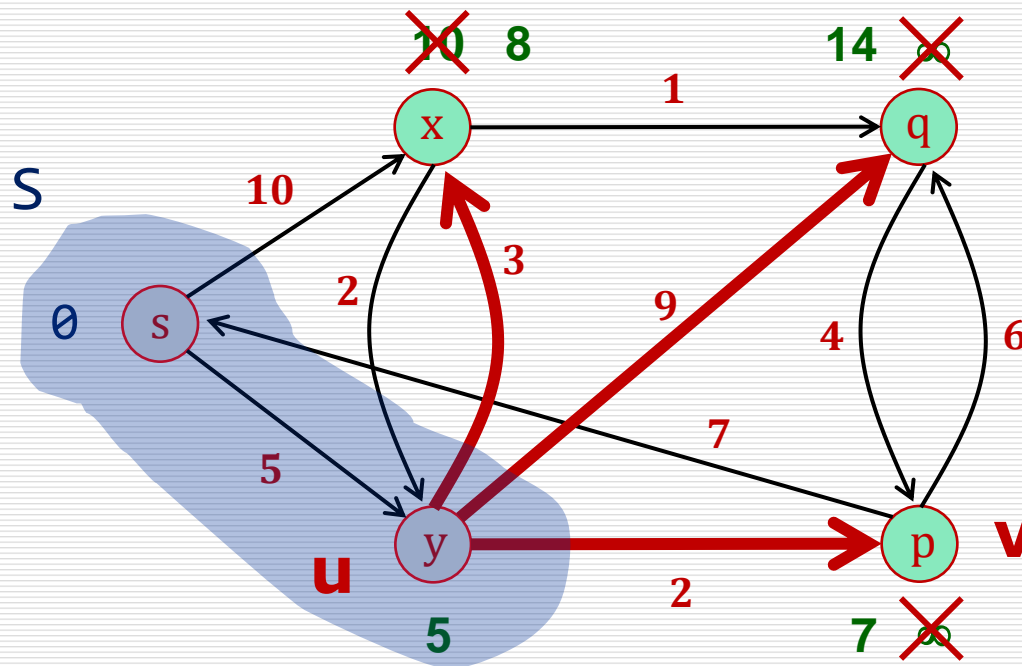
prev(y)=s

prev(p)=NIL

prev(q)=y

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, y, p, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

$S \leftarrow S \cup \{u\}$

για κάθε κόμβο  $v \in N(u)$

Update( $u, v, c$ )

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=13$

prev(s)=NIL

prev(x)=y

prev(y)=s

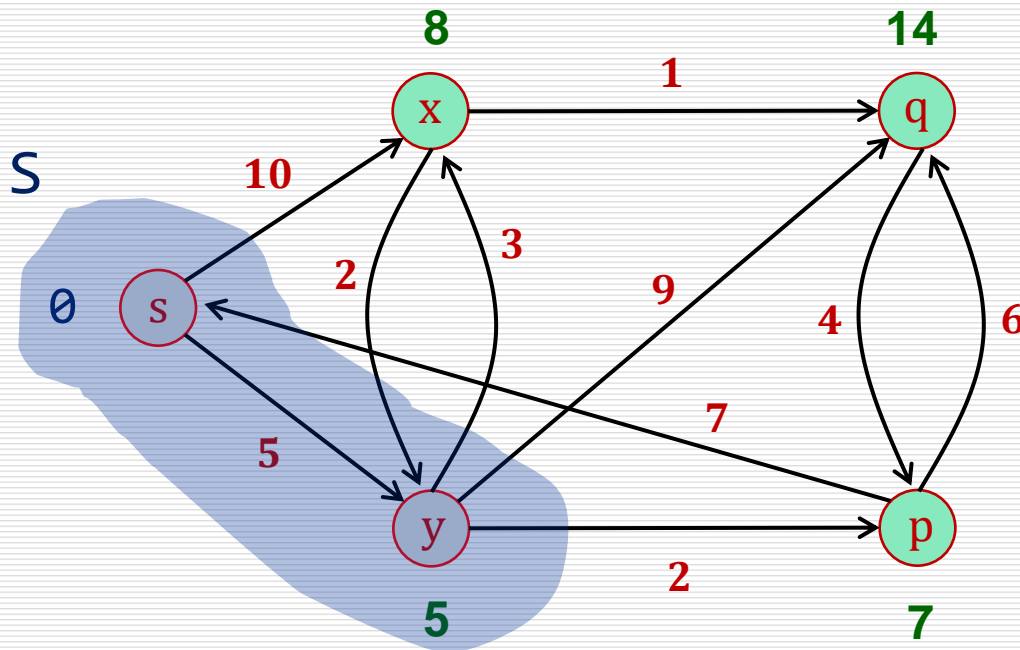
prev(p)=y

prev(q)=y

# 3ο παράδειγμα Dijkstra

▣ Παράδειγμα

```
Q = {x, p, q}
while Q ≠ ∅:
    u ← extract-min(Q);
```

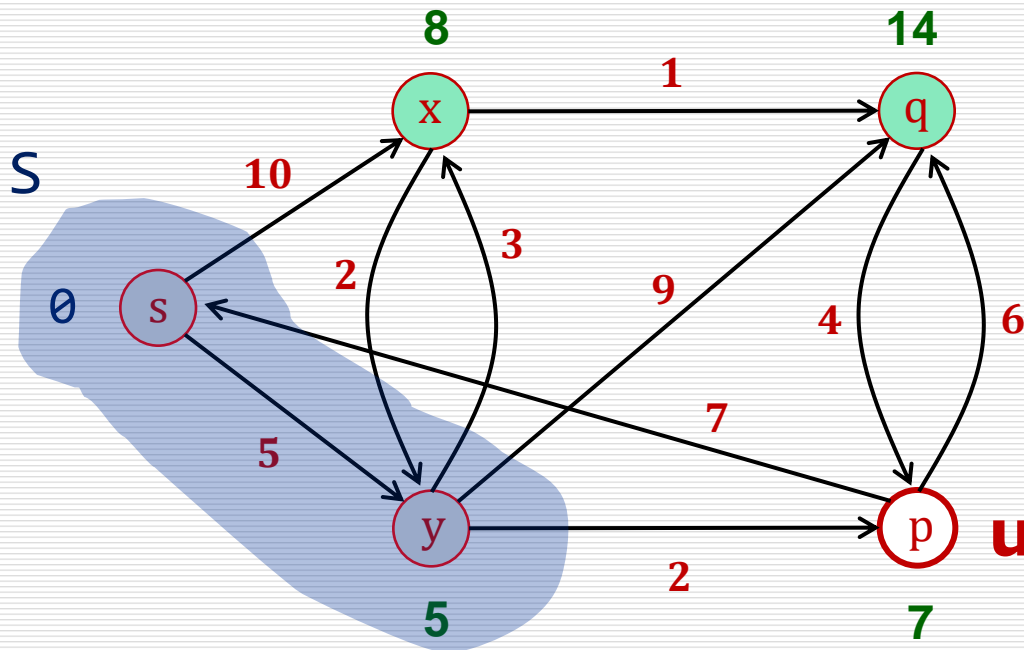


---

$d(s)=0$	$d(x)=8$	$d(y)=5$	$d(p)=7$	$d(q)=13$
$\text{prev}(s)=\text{NIL}$	$\text{prev}(x)=y$	$\text{prev}(y)=s$	$\text{prev}(p)=y$	$\text{prev}(q)=y$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, q\}$

while  $Q \neq \emptyset$ :

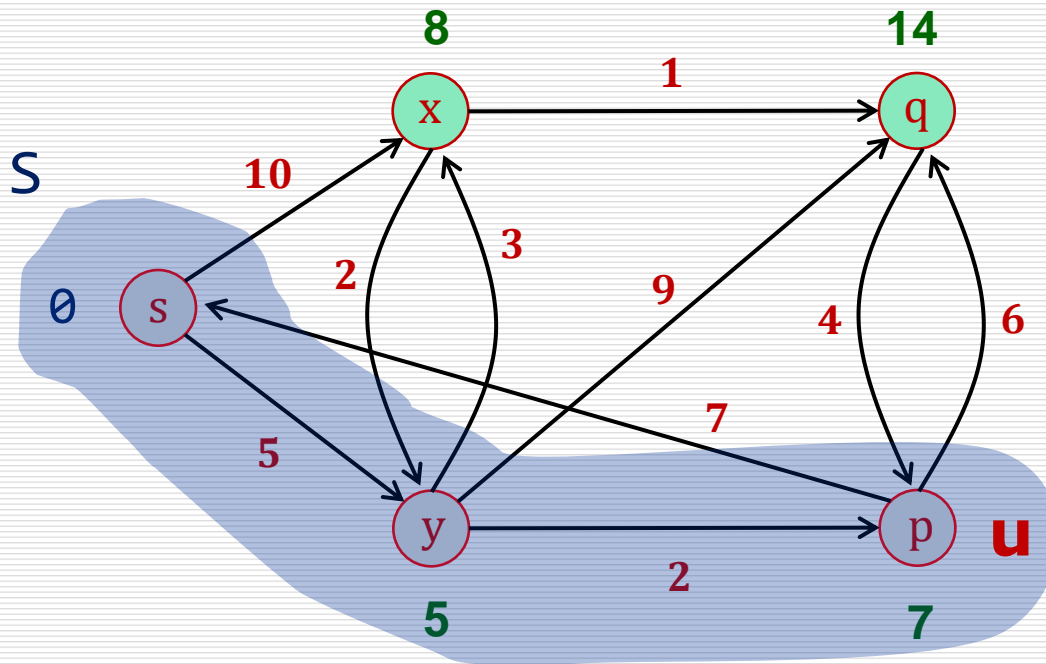
$u \leftarrow \text{extract-min}(Q);$

---

$d(s)=0$	$d(x)=8$	$d(y)=5$	$d(p)=7$	$d(q)=13$
$\text{prev}(s)=\text{NIL}$	$\text{prev}(x)=y$	$\text{prev}(y)=s$	$\text{prev}(p)=y$	$\text{prev}(q)=y$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$$Q = \{x, q\}$$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;

$S \leftarrow S \cup \{u\}$

---

$$d(s) = 0$$

$$\text{prev}(s) = \text{NIL}$$

$$d(x) = 8$$

$$\text{prev}(x) = y$$

$$d(y) = 5$$

$$\text{prev}(y) = s$$

$$d(p) = 7$$

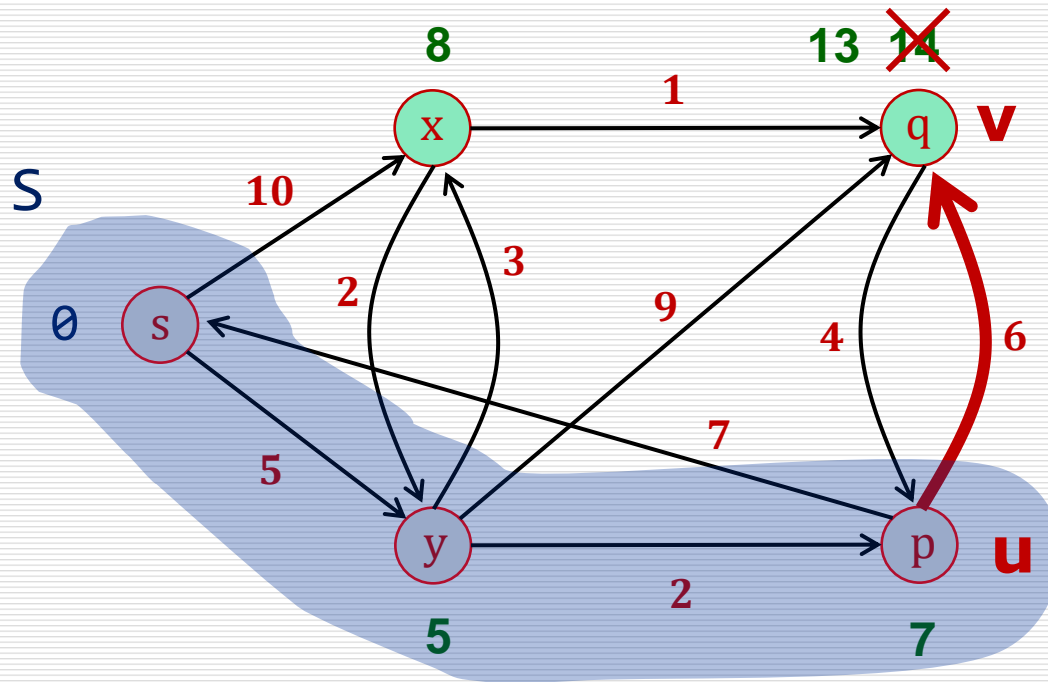
$$\text{prev}(p) = y$$

$$d(q) = 13$$

$$\text{prev}(q) = y$$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{x, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;

$S \leftarrow S \cup \{u\}$

για κάθε κόμβο  $v \in N(u)$

Update( $u, v, c$ )

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=13$

prev( $s$ )=NIL

prev( $x$ )= $y$

prev( $y$ )= $s$

prev( $p$ )= $y$

prev( $q$ )= $y$

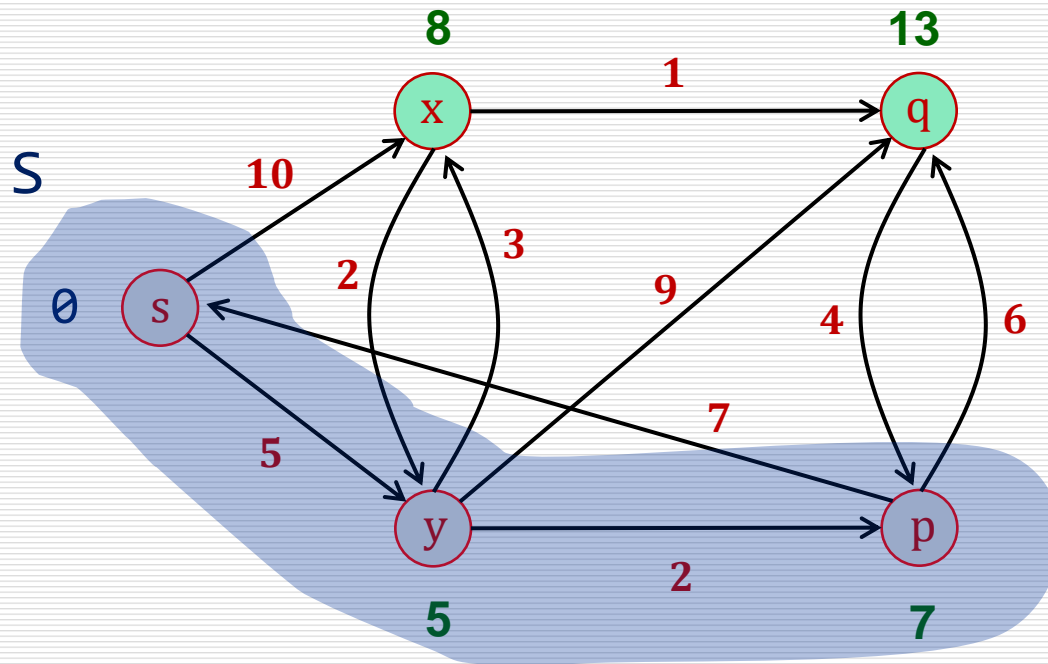
# 3ο παράδειγμα Dijkstra

## Παράδειγμα

$Q = \{x, q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;



$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=13$

$\text{prev}(s)=\text{NIL}$

$\text{prev}(x)=y$

$\text{prev}(y)=s$

$\text{prev}(p)=y$

$\text{prev}(q)=y$



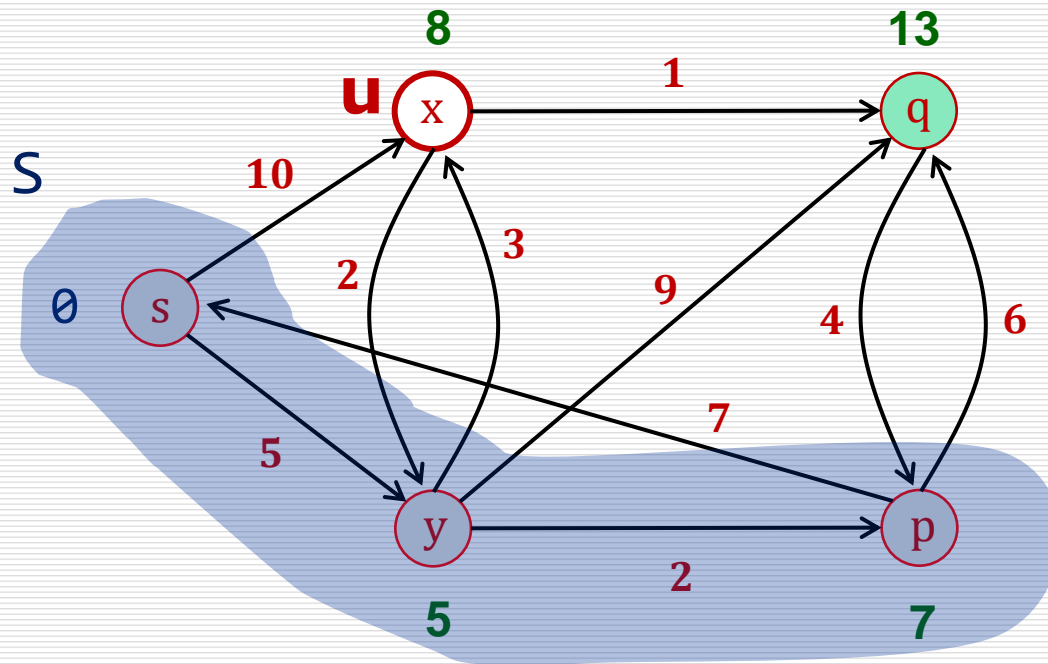
# 3ο παράδειγμα Dijkstra

■ Παράδειγμα

$Q = \{q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$



$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=13$

$\text{prev}(s)=\text{NIL}$

$\text{prev}(x)=y$

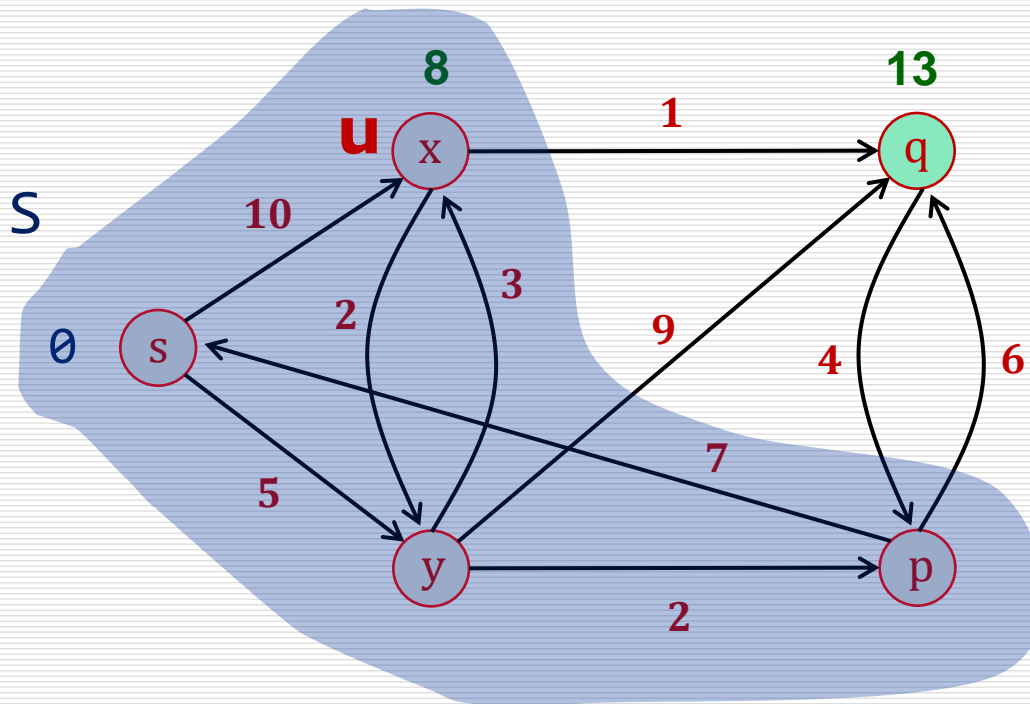
$\text{prev}(y)=s$

$\text{prev}(p)=y$

$\text{prev}(q)=y$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;

$S \leftarrow S \cup \{u\}$

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=13$

$\text{prev}(s)=\text{NIL}$

$\text{prev}(x)=y$

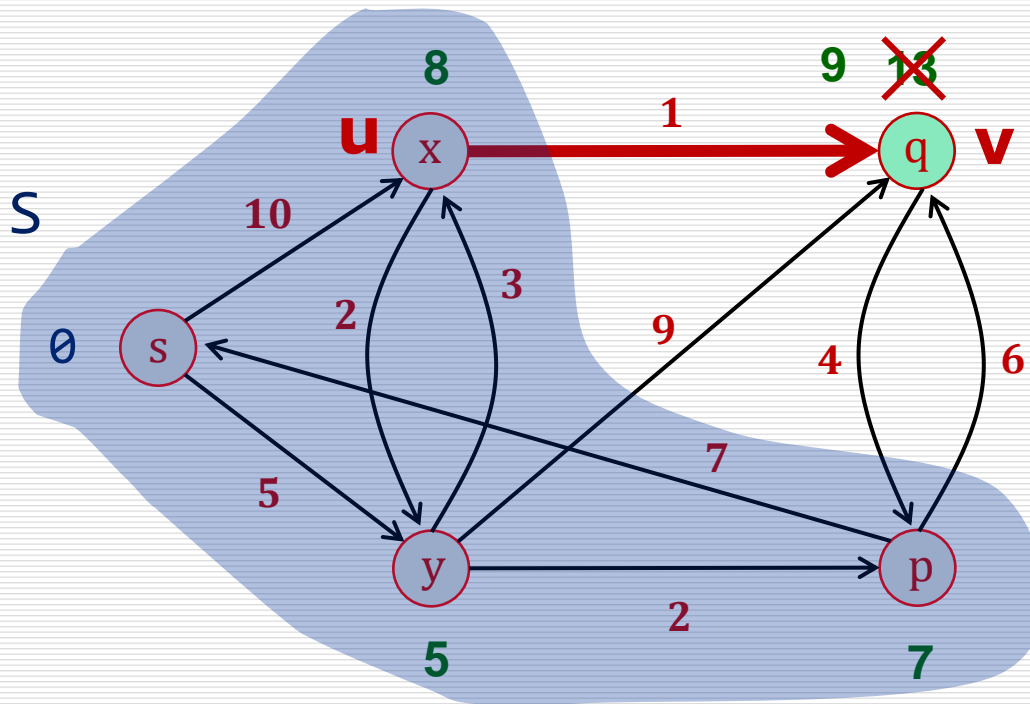
$\text{prev}(y)=s$

$\text{prev}(p)=y$

$\text{prev}(q)=y$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;

$S \leftarrow S \cup \{u\}$

για κάθε κόμβο  $v \in N(u)$

Update( $u, v, c$ )

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=9$

prev( $s$ )=NIL

prev( $x$ )= $y$

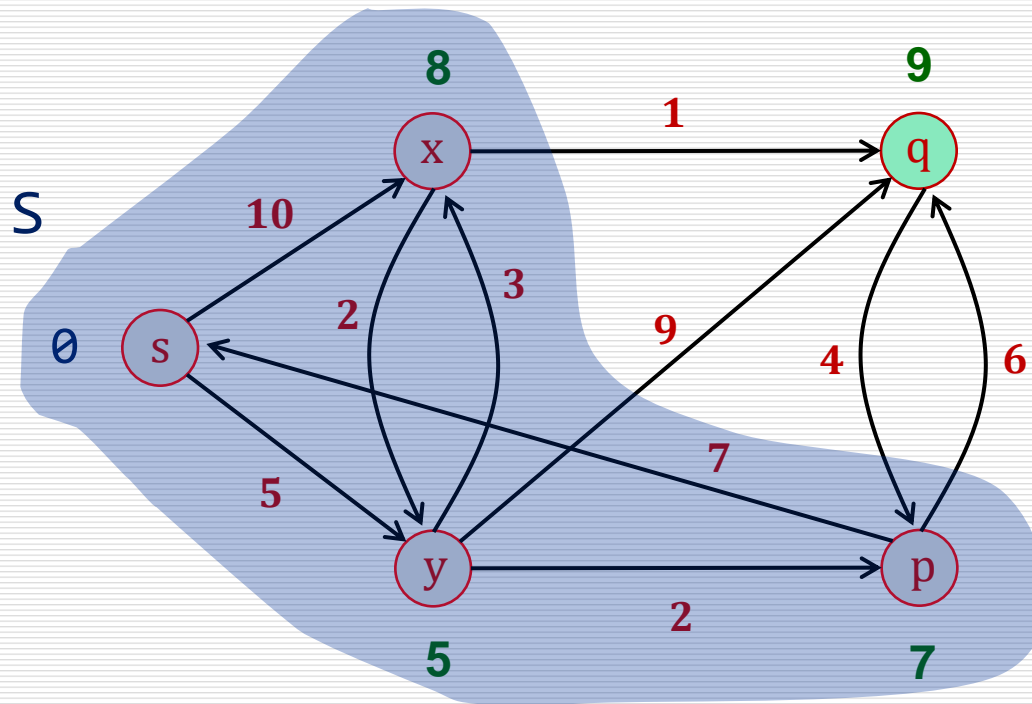
prev( $y$ )= $s$

prev( $p$ )= $y$

prev( $q$ )= $x$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{q\}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q);$

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=9$

$\text{prev}(s)=\text{NIL}$

$\text{prev}(x)=y$

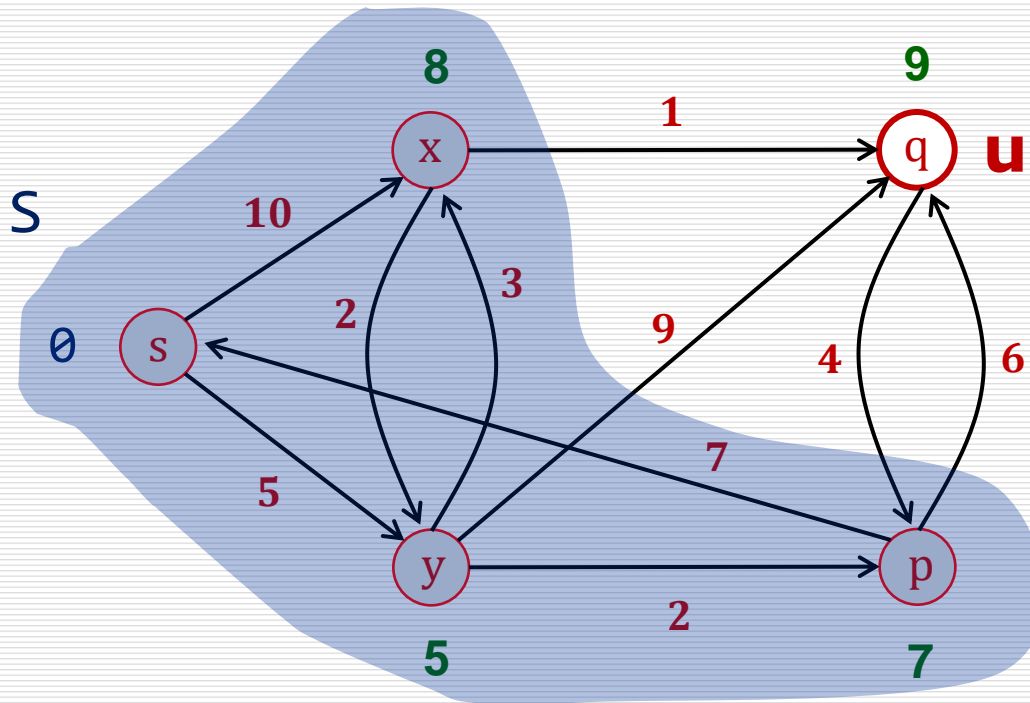
$\text{prev}(y)=s$

$\text{prev}(p)=y$

$\text{prev}(q)=x$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{ \}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=9$

$\text{prev}(s)=\text{NIL}$

$\text{prev}(x)=y$

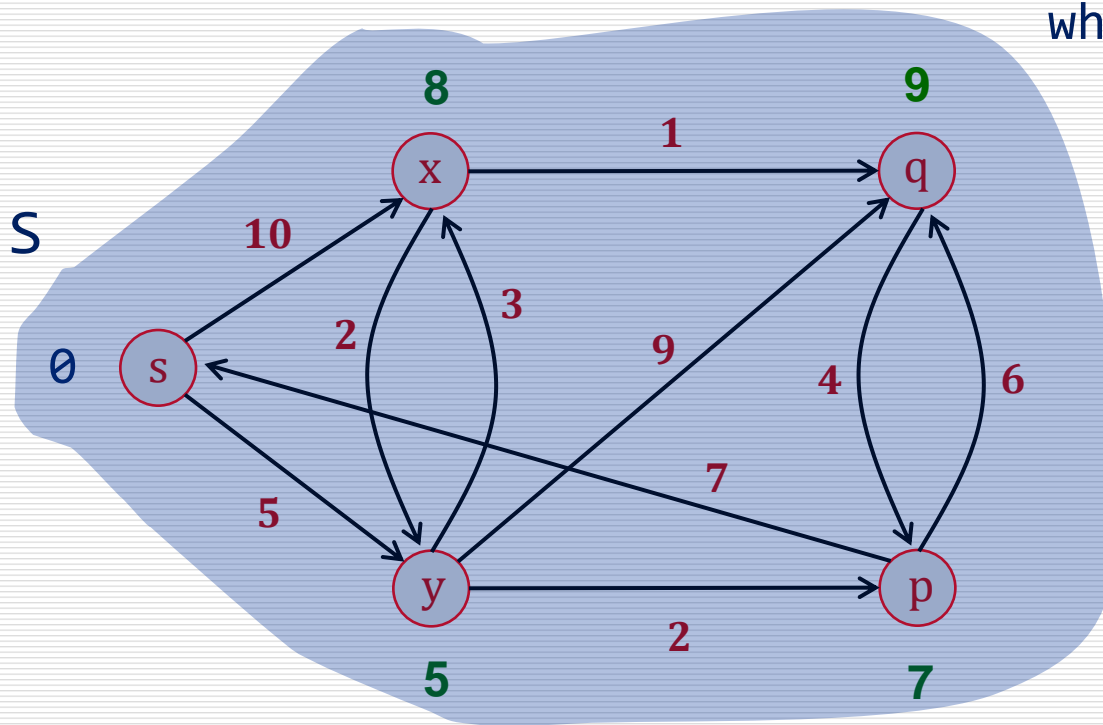
$\text{prev}(y)=s$

$\text{prev}(p)=y$

$\text{prev}(q)=x$

# 3ο παράδειγμα Dijkstra

## Παράδειγμα



$Q = \{ \}$

while  $Q \neq \emptyset$ :

$u \leftarrow \text{extract-min}(Q)$ ;

$S \leftarrow S \cup \{u\}$

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=9$

$\text{prev}(s)=\text{NIL}$

$\text{prev}(x)=y$

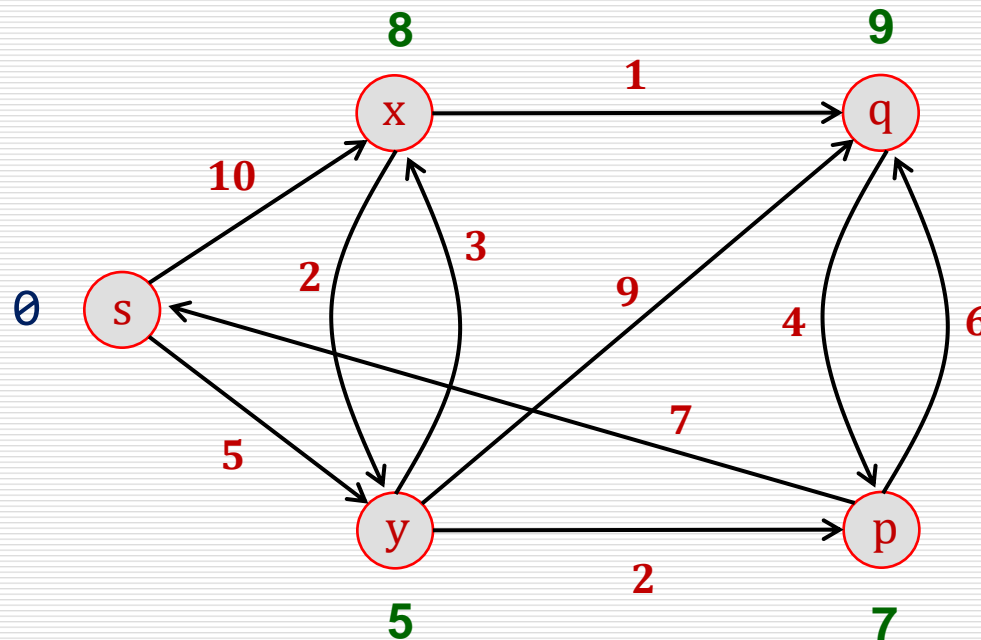
$\text{prev}(y)=s$

$\text{prev}(p)=y$

$\text{prev}(q)=x$

# 3ο παράδειγμα Dijkstra

▣ Παράδειγμα



$Q = \{ \}$

while  $Q \neq \emptyset$ :

return  $d()$ ,  $prev()$

$d(s)=0$

$d(x)=8$

$d(y)=5$

$d(p)=7$

$d(q)=9$

$prev(s)=NIL$

$prev(x)=y$

$prev(y)=s$

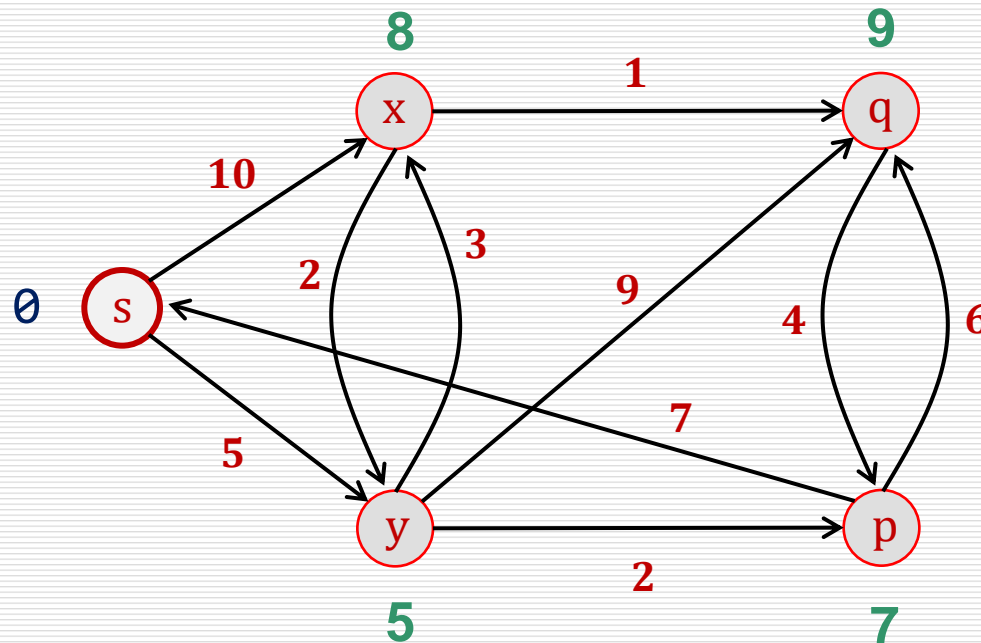
$prev(p)=y$

$prev(q)=x$

# 3ο παράδειγμα Dijkstra

Παράδειγμα

Έξοδος Αλγόριθμου



Κόστος  
ελάχιστων  
διαδρομών

$$d(s)=0$$

$$\text{prev}(s)=\text{NIL}$$

$$d(x)=8$$

$$\text{prev}(x)=y$$

$$d(y)=5$$

$$\text{prev}(y)=s$$

$$d(p)=7$$

$$\text{prev}(p)=y$$

$$d(q)=9$$

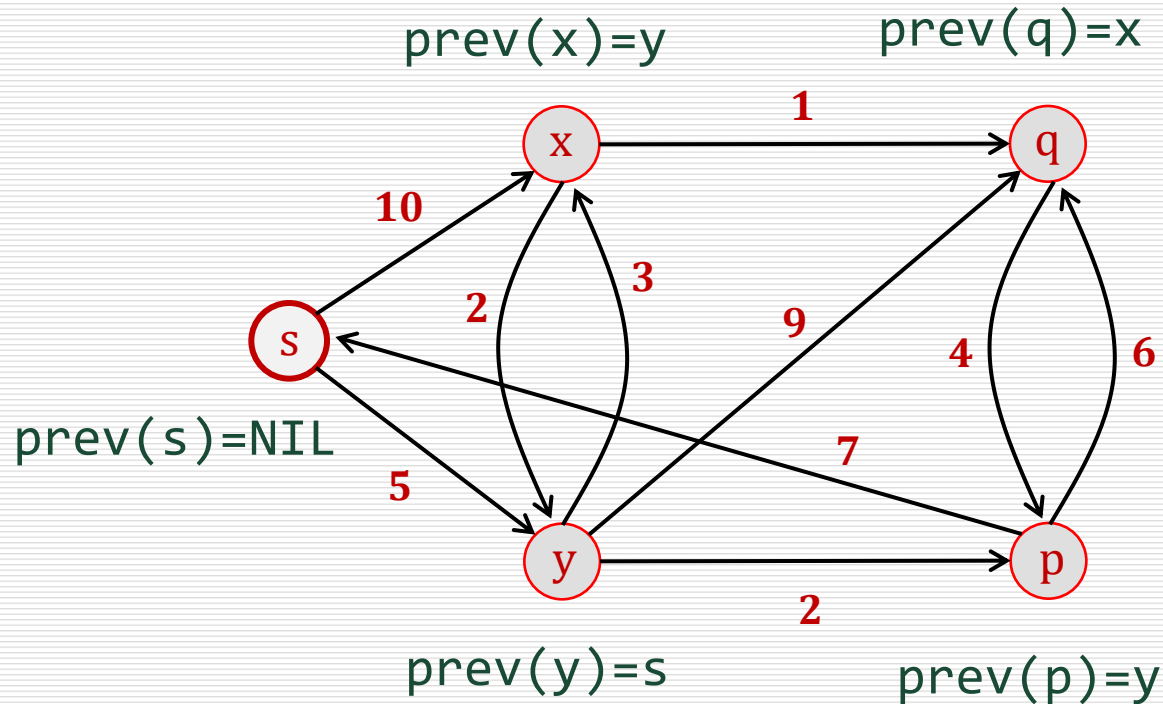
$$\text{prev}(q)=x$$



# 3ο παράδειγμα Dijkstra

■ Παράδειγμα

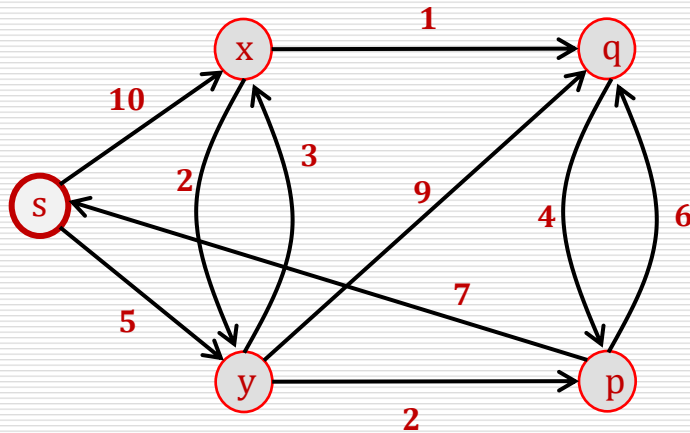
Έξοδος Αλγόριθμου



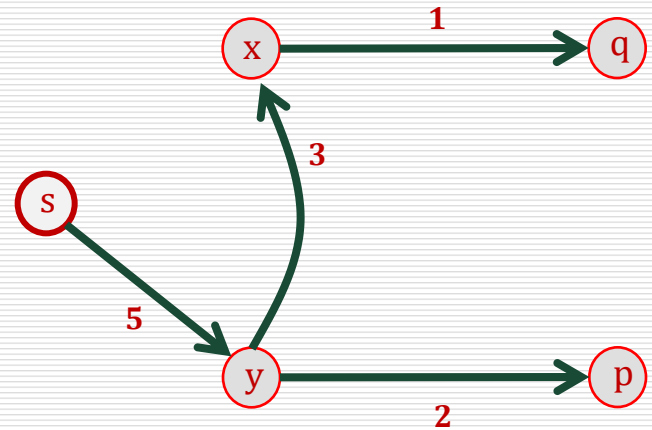
**Προηγούμενοι**  
κόμβοι στις  
ελάχιστες  
διαδρομές

# 3ο παράδειγμα Dijkstra

## Παράδειγμα

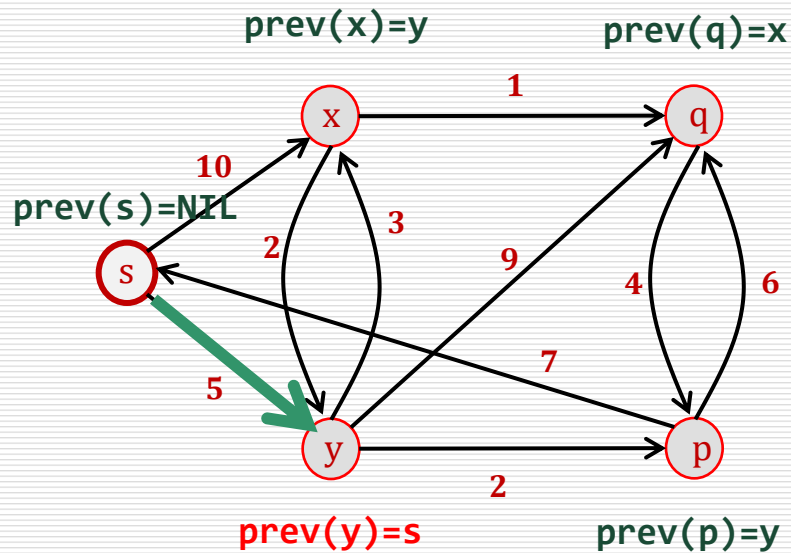


## Δένδρο Ελάχιστων Διαδρομών

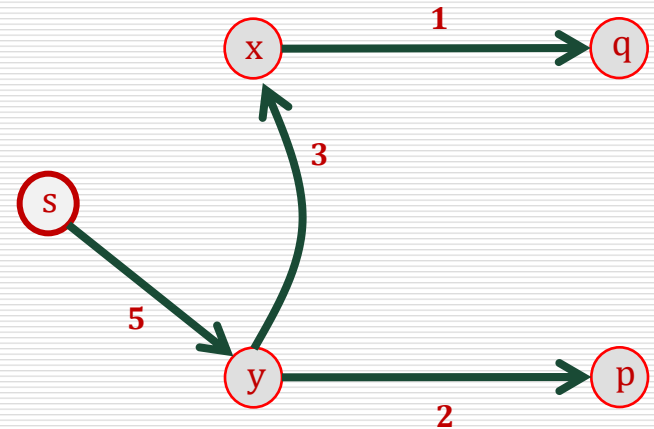


# 3ο παράδειγμα Dijkstra

## Παράδειγμα

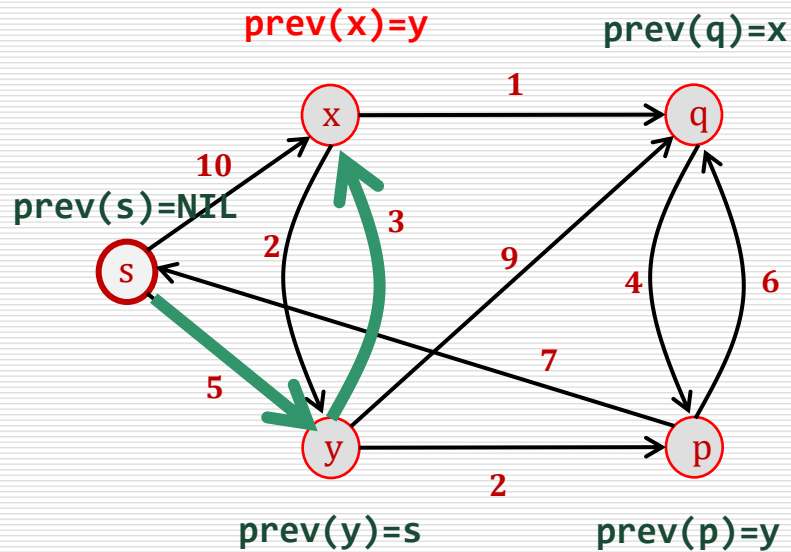


## Δένδρο Ελάχιστων Διαδρομών ΚΑΤΑΣΚΕΥΗ

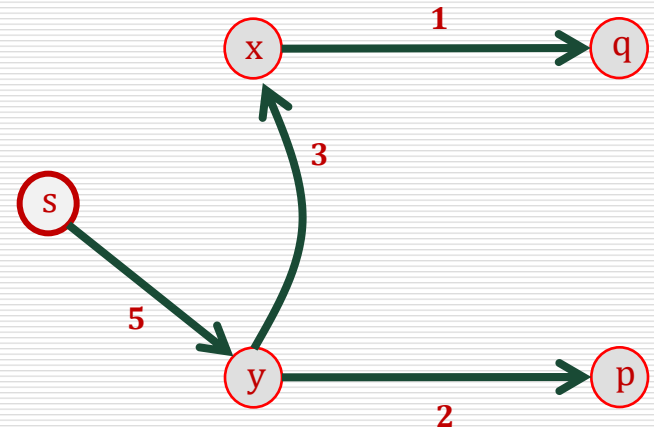


# 3ο παράδειγμα Dijkstra

## Παράδειγμα

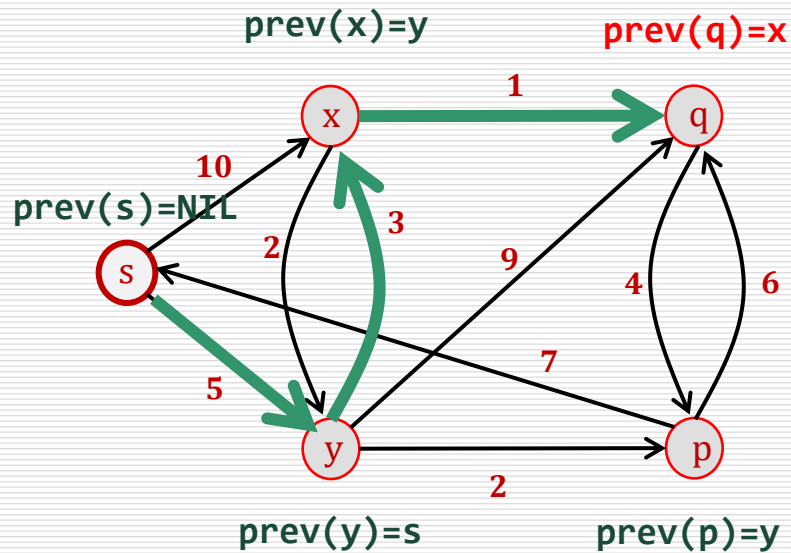


## Δένδρο Ελάχιστων Διαδρομών ΚΑΤΑΣΚΕΥΗ

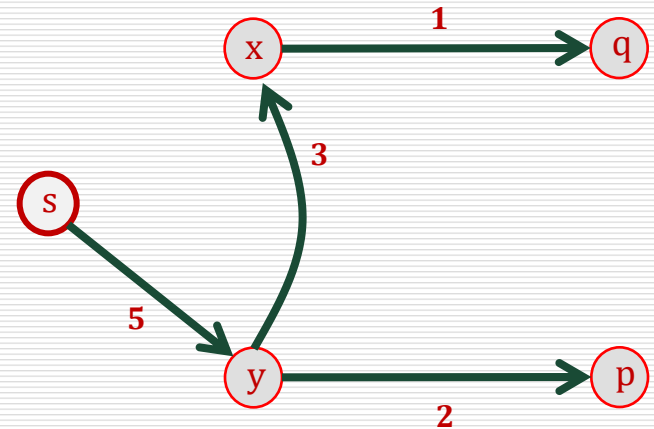


# 3ο παράδειγμα Dijkstra

## Παράδειγμα

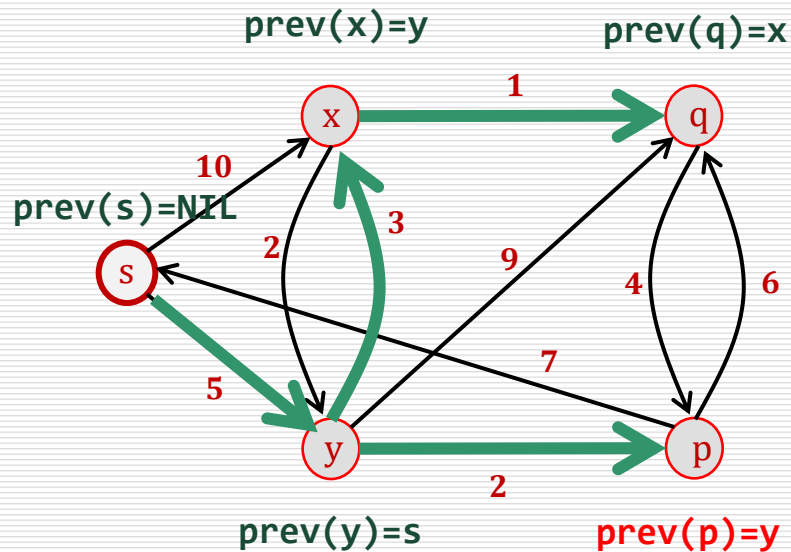


## Δένδρο Ελάχιστων Διαδρομών ΚΑΤΑΣΚΕΥΗ

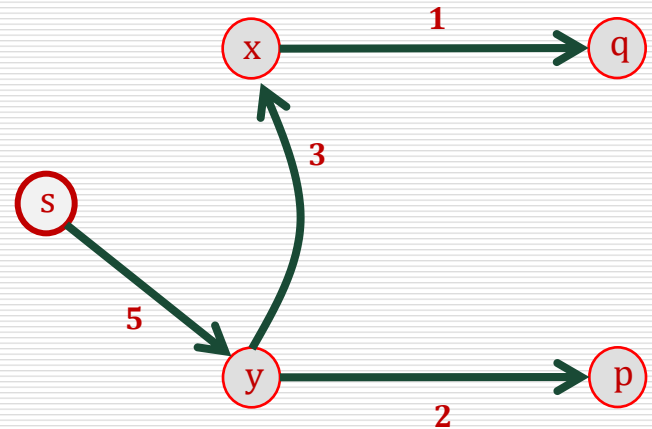


# 3ο παράδειγμα Dijkstra

## Παράδειγμα

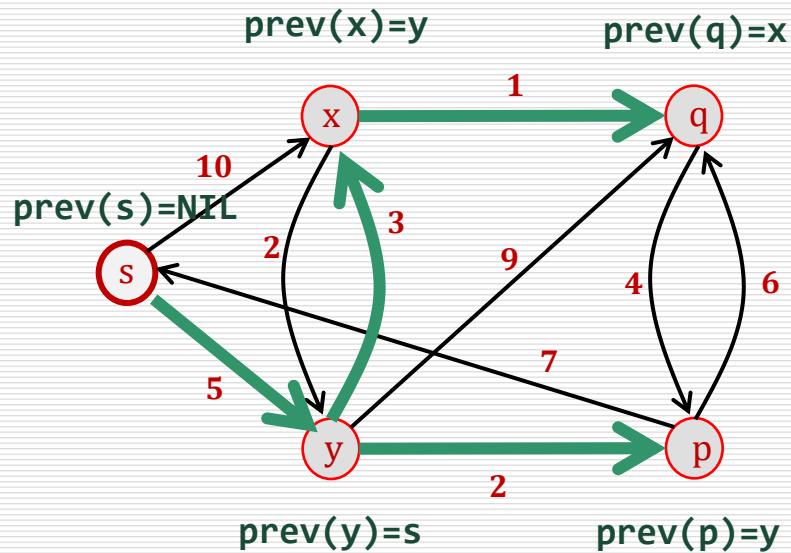


## Δένδρο Ελάχιστων Διαδρομών ΚΑΤΑΣΚΕΥΗ

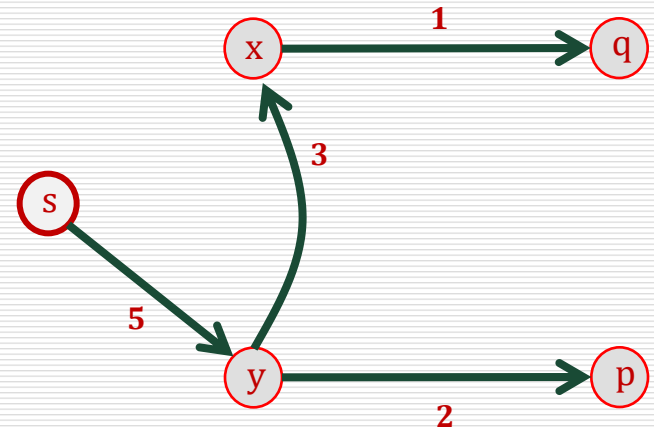


# 3ο παράδειγμα Dijkstra

## Παράδειγμα

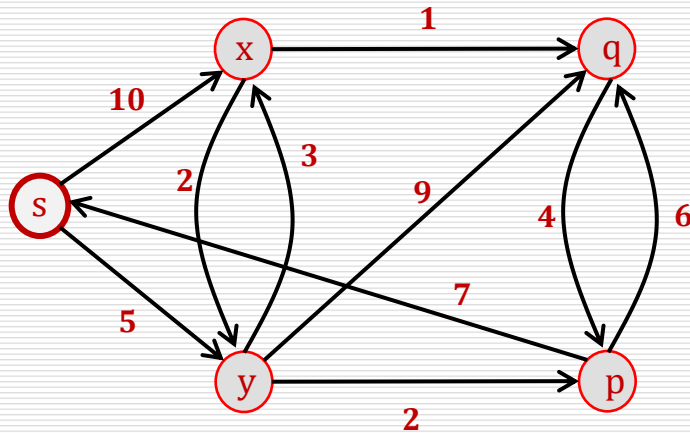


## Δένδρο Ελάχιστων Διαδρομών ΚΑΤΑΣΚΕΥΗ

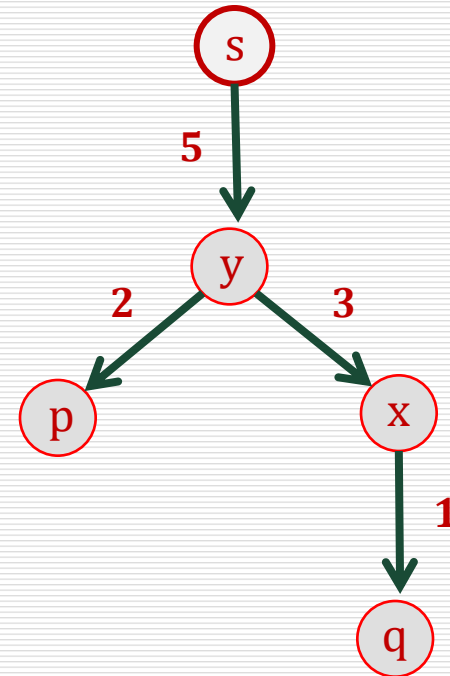


# 3ο παράδειγμα Dijkstra

## Παράδειγμα



Δένδρο Ελάχιστων Διαδρομών  
ΚΑΤΑΣΚΕΥΗ





# Πολυπλοκότητα Dijkstra

---

- Ομοιότητες με αλγόριθμο BFS;
  - Πιο αργός από BFS (ουρά προτεραιότητας vs απλή ουρά)
  - Απαιτεί  $n = |V|$  λειτουργίες insert στην ουρά, και  $m = |E|$  λειτουργίες update:
    - $|V|$  insert / extract-min
    - $|E|$  update
  - Υλοποίηση ουράς με πίνακα  $\Rightarrow O(|V|^2)$
  - Υλοποίηση ουράς με δυαδικό σωρό  $\Rightarrow O(|E| \log |V|)$
  - Υλοποίηση ουράς με σωρό Fibonacci  $\Rightarrow O(|E| + |V| \log |V|)$
-

# Ορθότητα αλγορίθμου Dijkstra

---

- Ο αλγόριθμος δημιουργεί σταδιακά ένα **δένδρο συντομότερων μονοπατιών**. Το δένδρο αρχικοποιείται με τον αρχικό κόμβο  $s$ .
- Σε κάθε επανάληψη επιλέγεται ο κόμβος  $w$  με την **ελάχιστη τρέχουσα απόσταση από τον  $s$** .
- Η απόδειξη στηρίζεται σε δύο αναλλοίωτες συνθήκες βρόχου.

# Ορθότητα αλγορίθμου Dijkstra

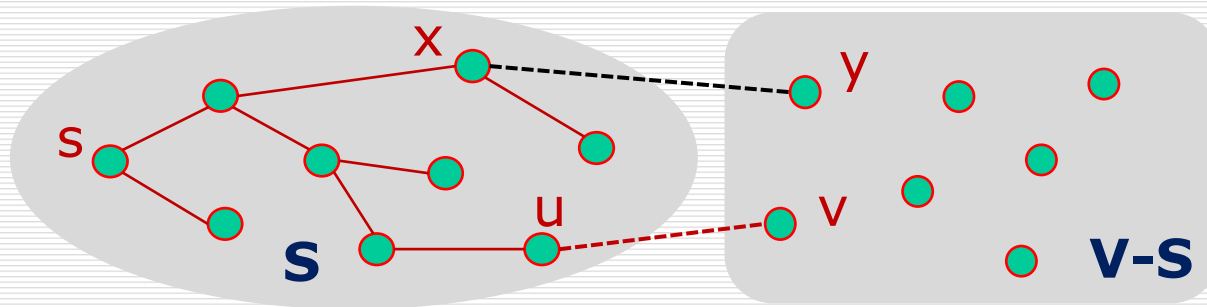
---

- **1η αναλλοίωτη βρόχου:** πριν από κάθε επανάληψη του εξωτερικού βρόχου, **τρέχουσα απόσταση** (κάθε) **κόμβου**  $d(v)$  = κόστος ελάχιστης διαδρομής από τον  $s$  στον  $v$ , μεταξύ όλων των διαδρομών που περνούν μόνο από κόμβους του συνόλου  $S$  (δηλ. του μέχρι στιγμής δένδρου).
- **2η αναλλοίωτη βρόχου:** πριν από κάθε επανάληψη του εξωτερικού βρόχου, για τον κόμβο  $w$  που ανήκει στο  $V-S$  και έχει **ελάχιστη** (μεταξύ κόμβων του  $V-S$ ) **τρέχουσα απόσταση από τον  $s$** , η απόσταση αυτή είναι η **τελική** (ελάχιστη) του  $w$  από τον  $s$ .
- Οι δύο αναλλοίωτες αποδεικνύονται **μαζί**, χρησιμοποιώντας επαγωγή.

# Ορθότητα αλγορίθμου Dijkstra

## Λήμμα (βήμα επαγωγής)

Έστω  $G = (V, E)$  γράφος με μη-αρνητικά βάρη,  $S \in V$ , και  $(S, V-S)$  μια διαμέριση του  $V$  τ.ώ.  $s \in S$  και γνωρίζουμε ελάχιστη απόσταση  $d(s, u) \forall u \in S$ .



Εάν  $(u, v)$  είναι η ακμή που ελαχιστοποιεί την ποσότητα

$$d(s, u) + c(u, v)$$

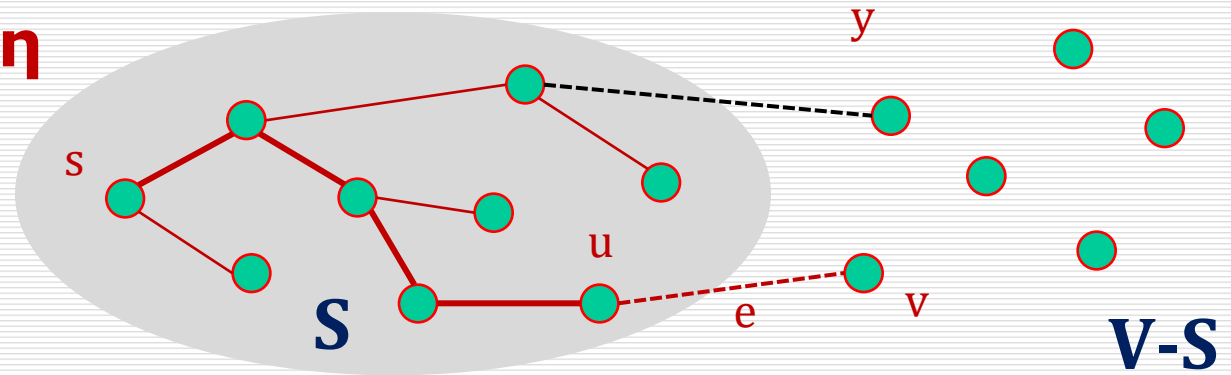
μεταξύ όλων των ακμών  $(x, y)$  με  $x \in S$  και  $y \in V-S$ , τότε

$$P = (s, \dots, u, v) \text{ είναι ε.δ. } s \rightsquigarrow v.$$

# Ορθότητα αλγορίθμου Dijkstra

---

## Απόδειξη



Έστω  $e = (u, v)$  και έστω  $(s, \dots, u)$  η ελάχιστη διαδρομή από τον  $s$  στον  $u$ .

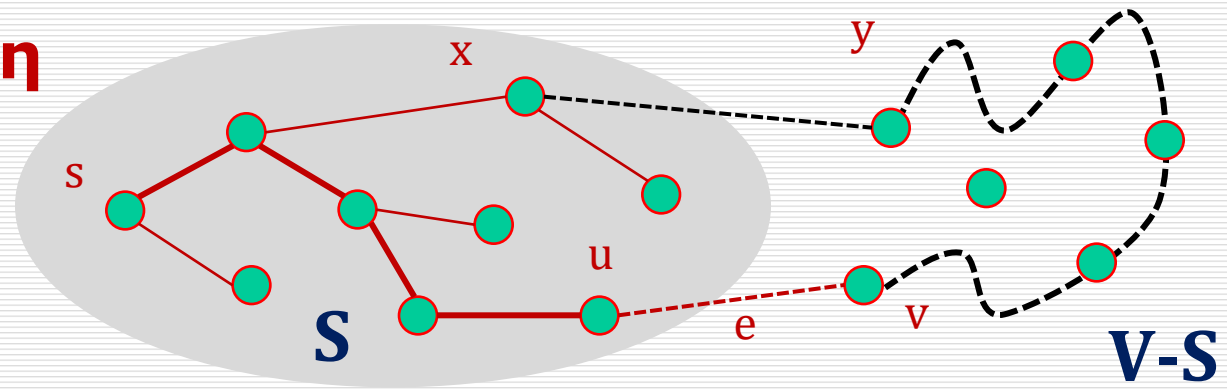
Για την διαδρομή  $P = (s, \dots, u, v)$  από τον  $s$  στον  $v$  ισχύει :

$$c(P) = d(s,u) + c(u,v) \quad (1)$$

# Ορθότητα αλγορίθμου Dijkstra

---

## Απόδειξη



Έστω  $Q = (s, \dots, x, y, \dots, v)$  μια ελάχιστη διαδρομή από τον  $s$  στον  $v$ , και

έστω  $y$  ο πρώτος κόμβος της διαδρομής  $Q : y \in V-S$

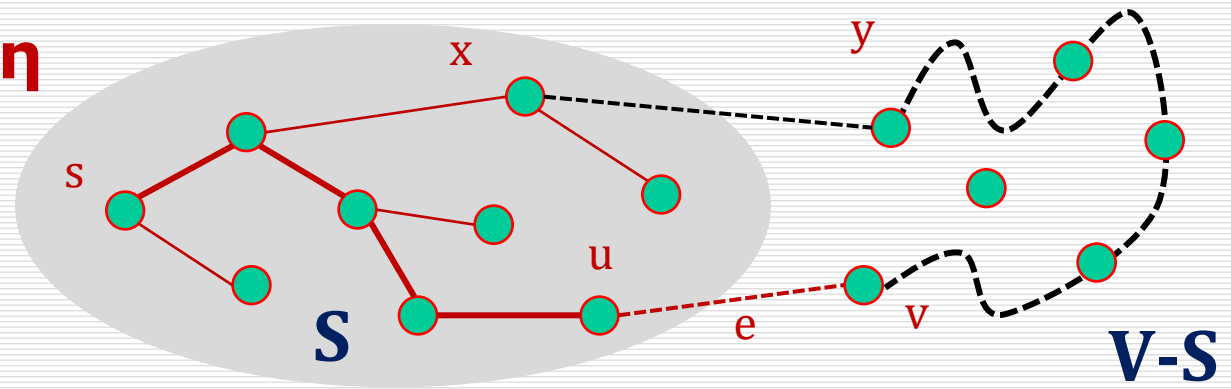
Θα δείξουμε ότι  $c(P) \leq c(Q)$

---

# Ορθότητα αλγορίθμου Dijkstra

---

## Απόδειξη



Από (1) και από επιλογή της ακμής  $e = (u, v)$ , έχουμε:

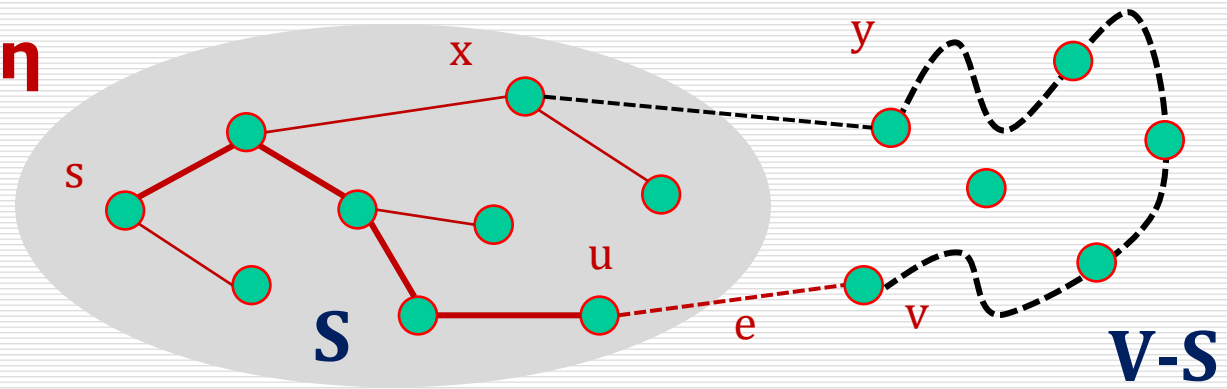
$$c(\mathbf{P}) = d(s, u) + c(u, v) \leq d(s, x) + c(x, y) \leq c(\mathbf{Q})$$

---

# Ορθότητα αλγορίθμου Dijkstra

---

## Απόδειξη



Από (1) και από επιλογή της ακμής  $e = (u, v)$ , έχουμε:

$$c(P) = d(s, u) + c(u, v) \leq d(s, x) + c(x, y) \leq c(Q)$$

Ερώτηση: πότε δεν ισχύει το παραπάνω;

---



# Ορθότητα αλγορίθμου Dijkstra

---

- Η ορθότητα ισχύει **μόνο** σε γράφους **χωρίς αρνητικά βάρη**
- *Άσκηση:* βρείτε αντιπαράδειγμα.

# Αλγόριθμος Bellman-Ford

---

```
dist(s) := 0 ;
```

```
for each  $v \neq s$  do dist(v) :=  $\infty$ 
```

```
repeat n-1 times
```

```
  for each edge  $e = (u,v)$  do
```

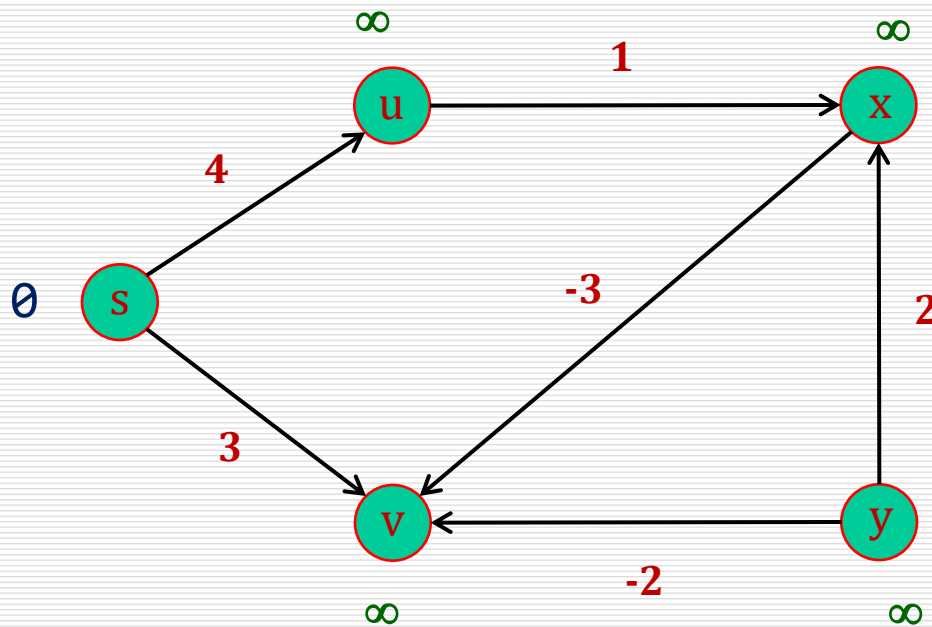
```
    if  $\text{dist}(u) + \text{cost}(u,v) < \text{dist}(v)$  then
```

```
       $\text{dist}(v) := \text{dist}(u) + \text{cost}(u,v)$ 
```

```
       $\text{prev}(v) := u$ 
```

# Αλγόριθμος Bellman-Ford

## Παραδείγματα



$$n = 5$$

$$m = 6$$

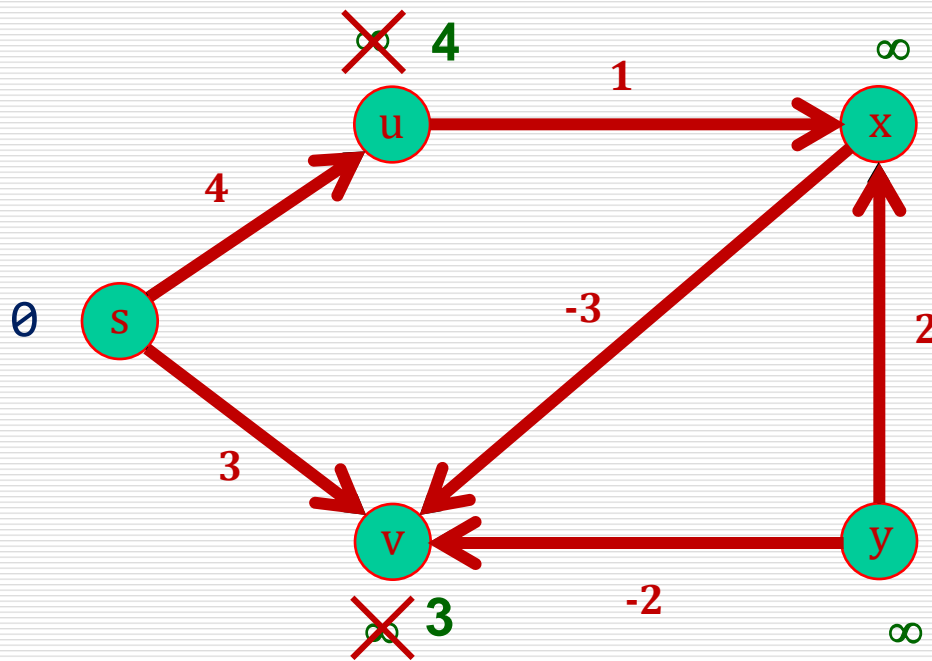
Update στις  
6 ακμές  
4 φορές

$$V = \{s, u, v, x, y\}$$

$$E = \{(y, x), (u, x), (y, v), (s, u), (x, v), (s, v)\}$$

# Αλγόριθμος Bellman-Ford

## Παράδειγμα 1



$n = 5$

$m = 6$

Update στις

6 ακμές

4 φορές

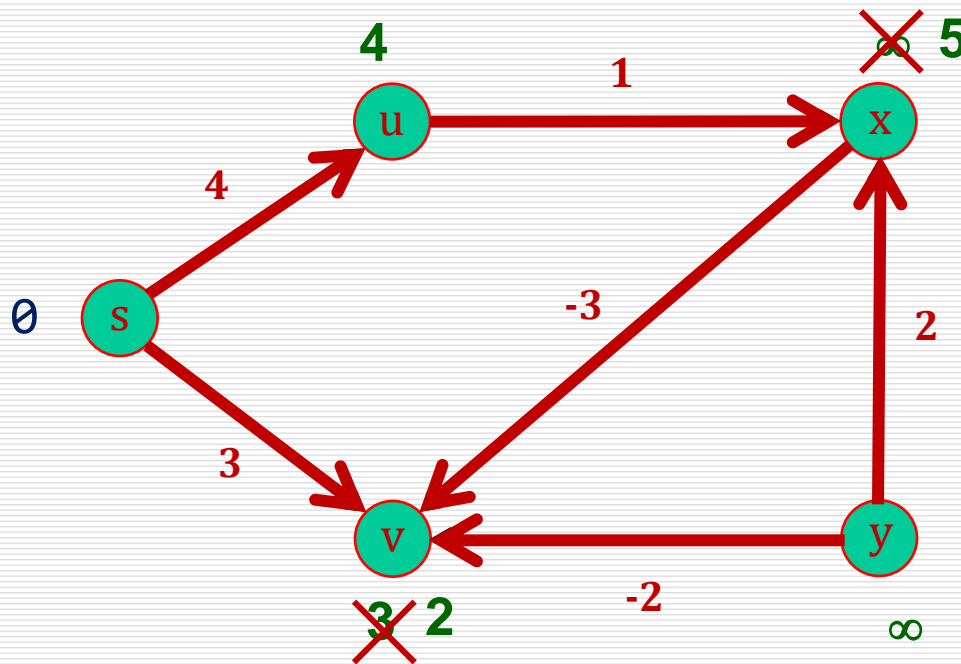
**1<sup>η</sup>** Επανάληψη

$S1: (y,x), (u,x), (y,v), (s,u), (x,v), (s,v)$



# Αλγόριθμος Bellman-Ford

## Παράδειγμα 1



$n = 5$

$m = 6$

Update στις

6 ακμές

4 φορές

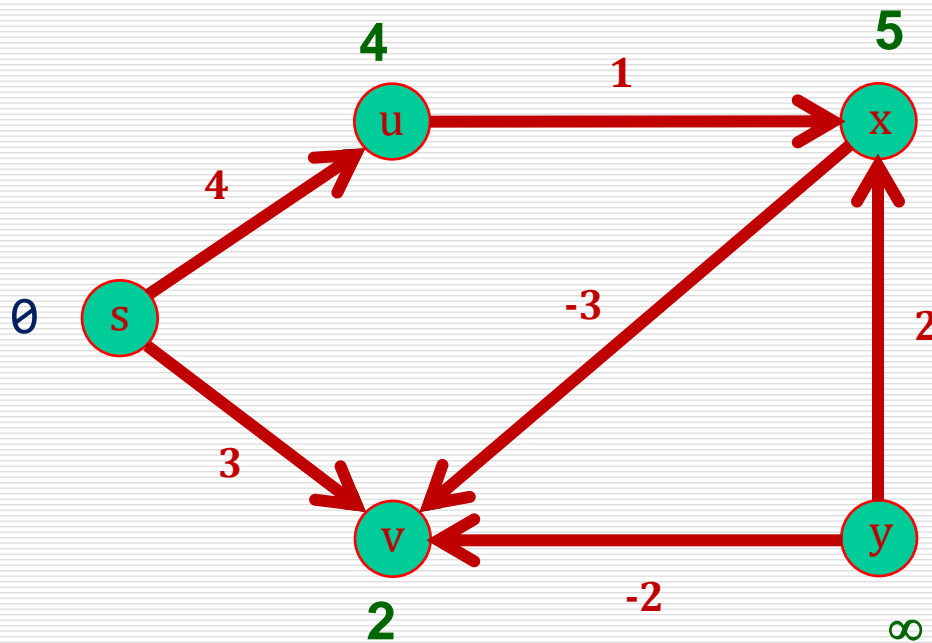
2<sup>η</sup> Επανάληψη

S1:  $(y,x), (u,x), (y,v), (s,u), (x,v), (s,v)$



# Αλγόριθμος Bellman-Ford

## Παράδειγμα 1



$n = 5$

$m = 6$

Update στις  
6 ακμές  
4 φορές

3<sup>η</sup> Επανάληψη

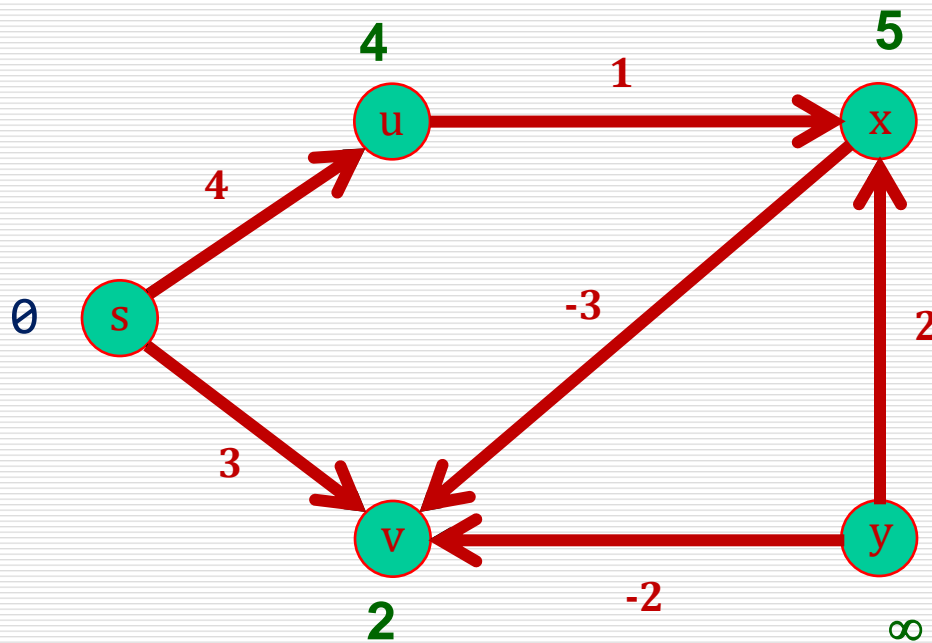
Καμία Αλλαγή  
Γιατί?

S1: (y,x), (u,x), (y,v), (s,u), (x,v), (s,v)



# Αλγόριθμος Bellman-Ford

## Παράδειγμα 1



$n = 5$

$m = 6$

Update στις  
6 ακμές  
4 φορές

4<sup>η</sup> Επανάληψη

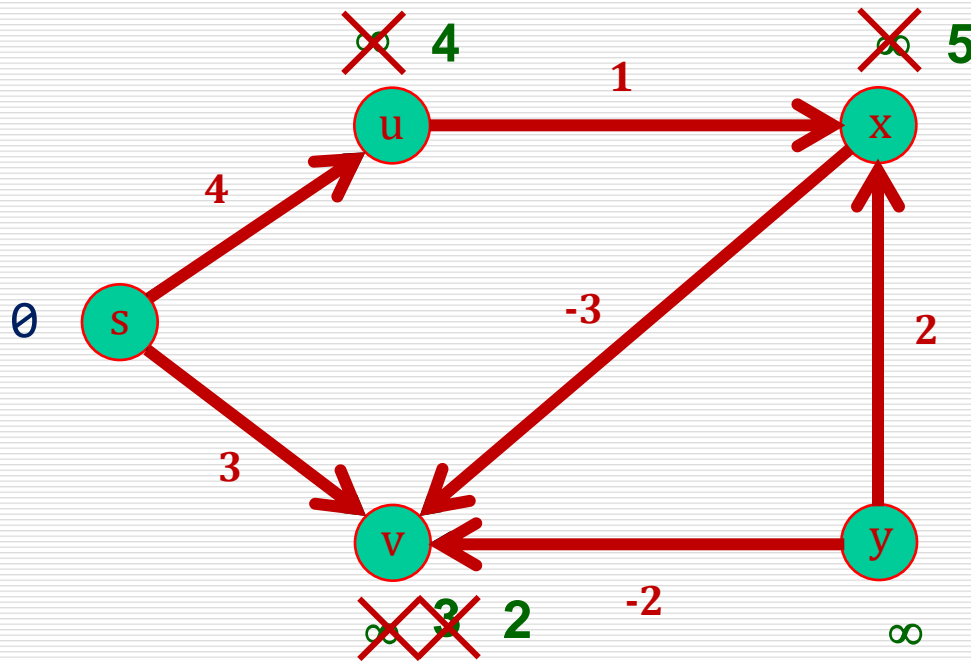
Καμία Αλλαγή

S1: (y,x), (u,x), (y,v), (s,u), (x,v), (s,v)



# Αλγόριθμος Bellman-Ford

Παράδειγμα 2



$n = 5$

$m = 6$

Update στις  
6 ακμές  
4 φορές

1<sup>η</sup> Επανάληψη

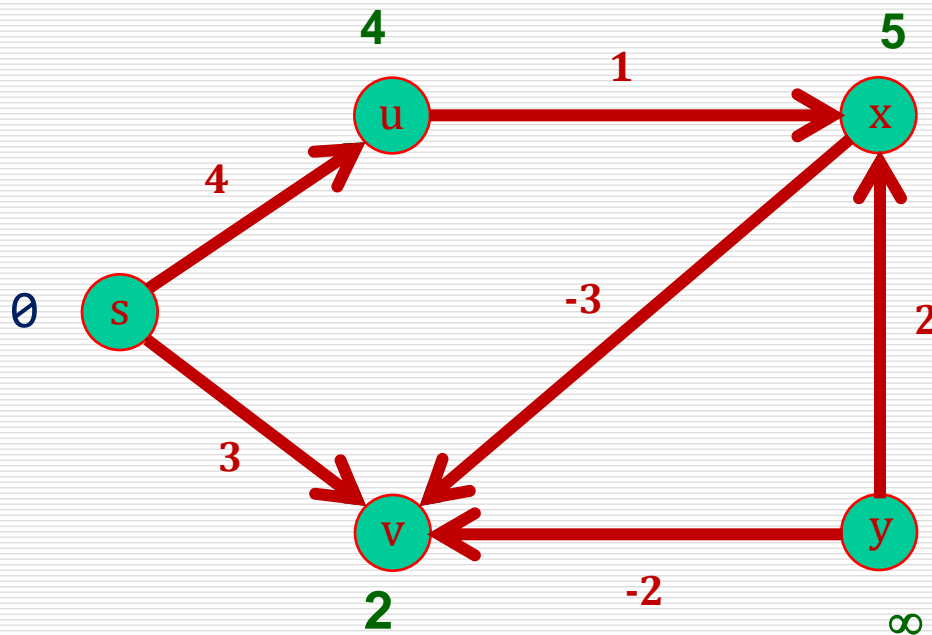
S2:  $(s, u)$ ,  $(u, x)$ ,  $(y, x)$ ,  $(s, v)$ ,  $(x, v)$ ,  $(y, v)$





# Αλγόριθμος Bellman-Ford

## Παράδειγμα 2



$n = 5$

$m = 6$

Update στις  
6 ακμές  
4 φορές

2<sup>η</sup> Επανάληψη

Καμία Αλλαγή  
Γιατί?

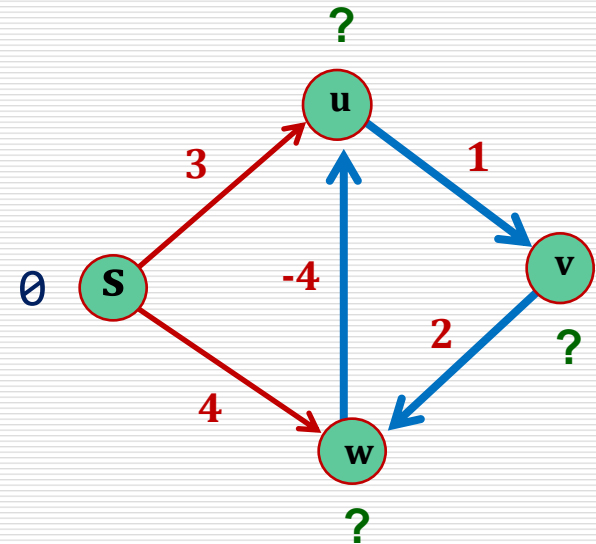
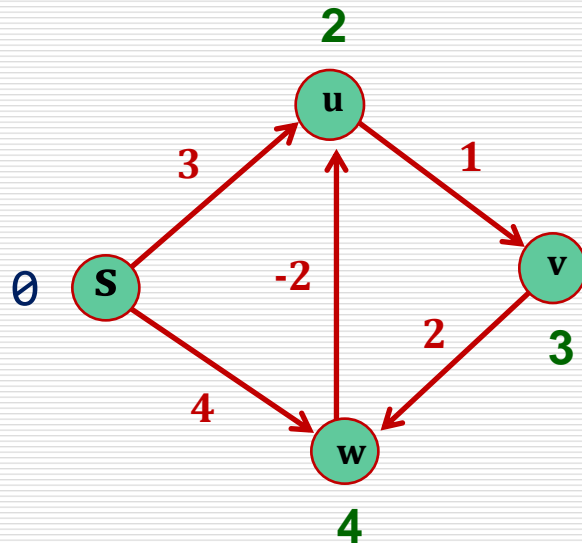
S2: (s,u), (u,x), (y,x), (s,v), (x,v), (y,v)



# Αλγόριθμος Bellman-Ford

---

## ● Αρνητικοί Κύκλοι



Εάν υπάρχει αρνητικός κύκλος δεν έχει νόημα να αναζητούμε ελάχιστες διαδρομές !!!

---

# Αλγόριθμος Bellman-Ford

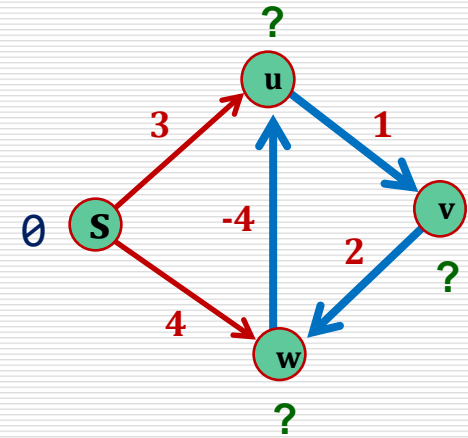
## Εντοπισμός αρνητικών κύκλων

Bellman-Ford-detection( $G, w, s$ )

1. initialize( $G, s$ )
2. επανάλαβε  $n-1$  φορές:  
για κάθε ακμή  $(u, v) \in E$ :  
Update( $u, v, c$ )

3. για κάθε ακμή  $(u, v) \in E$ :  
if  $d(v) > d(u) + c(u, v)$  then  
return FALSE

4. return  $d(\cdot)$



# Αλγόριθμος Bellman-Ford

---

- **Ορθότητα:** στο τέλος της  $k$ -οστής επανάληψης έχουν υπολογιστεί σωστά όλες οι ελάχιστες διαδρομές που χρησιμοποιούν το πολύ  $k$  ακμές (άσκηση: αποδείξτε το).
- **Πολυπλοκότητα:**  $O(|V||E|)$

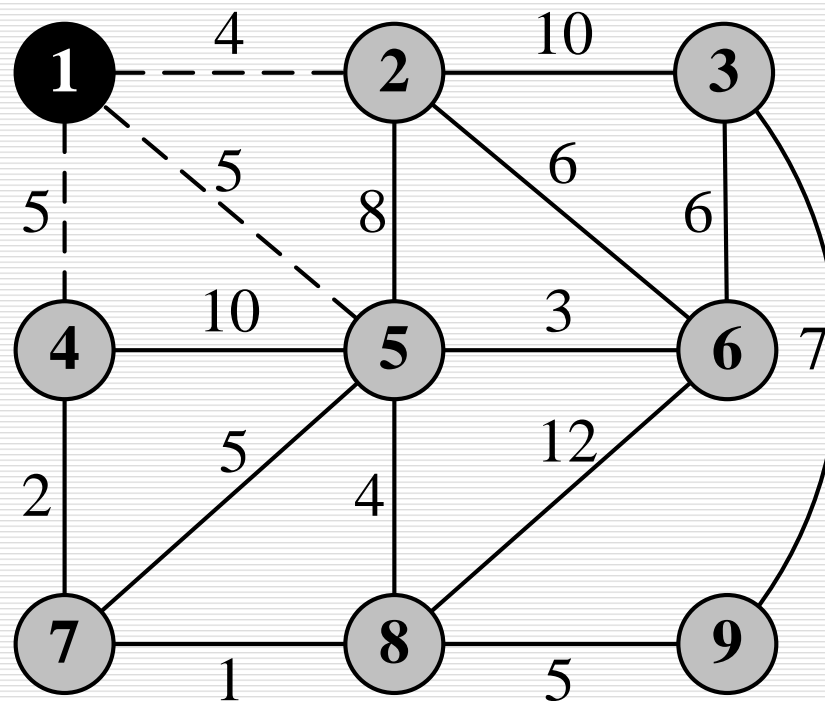
# Ελάχιστο Συνδετικό Δένδρο (MST)

---

- **Κριτήριο Prim:** Διαλέγουμε κάθε φορά την ακμή ελαχίστου κόστους έτσι ώστε ο νέος υπογράφος να παραμένει δένδρο (έναρξη από οποιονδήποτε κόμβο)
- **Κριτήριο Kruskal:** Διαλέγουμε κάθε φορά την ακμή ελαχίστου κόστους έτσι ώστε ο νέος υπογράφος να μην έχει κύκλους

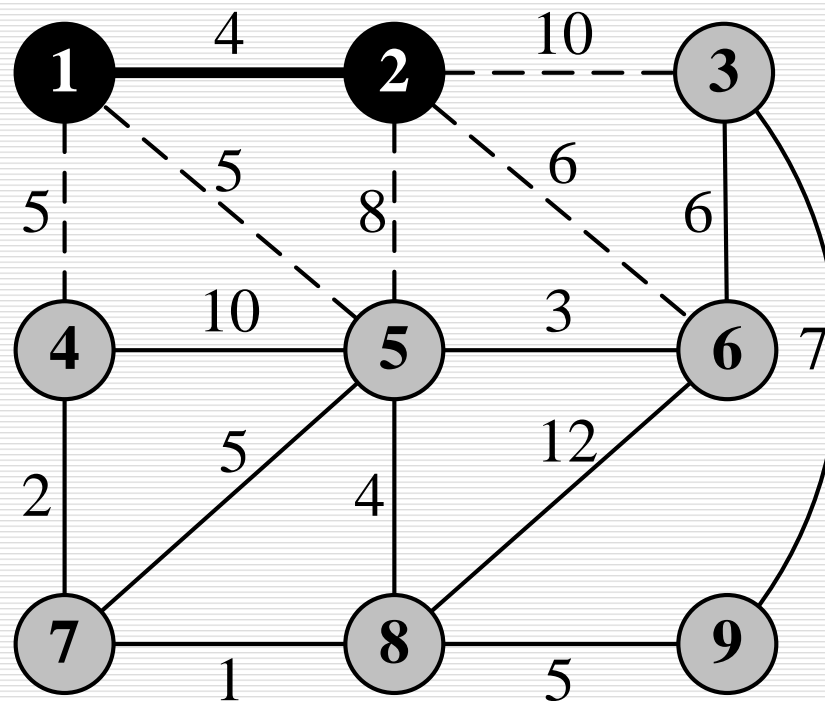
# Αλγόριθμος Prim: παράδειγμα

---



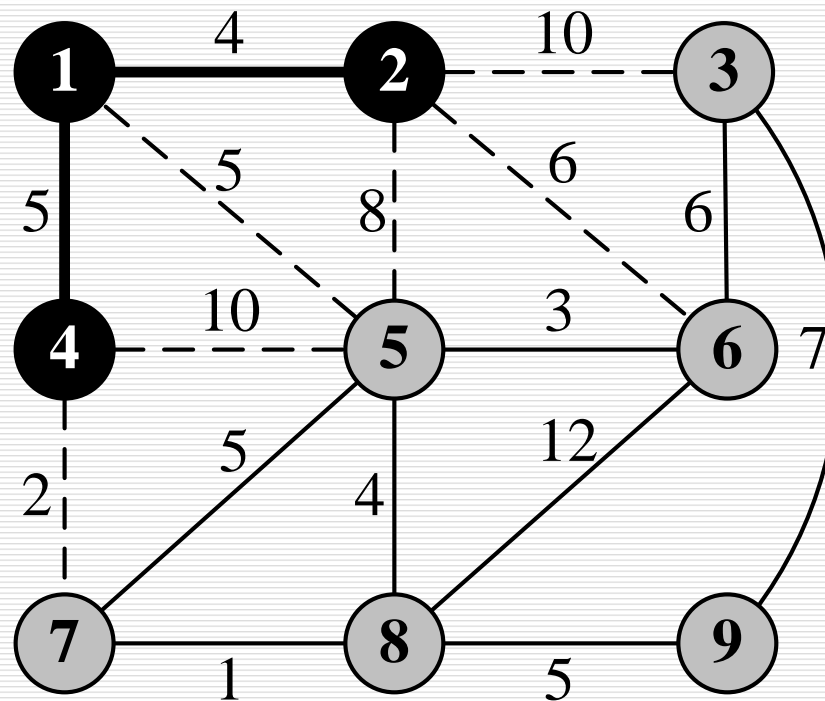
# Αλγόριθμος Prim: παράδειγμα

---



# Αλγόριθμος Prim: παράδειγμα

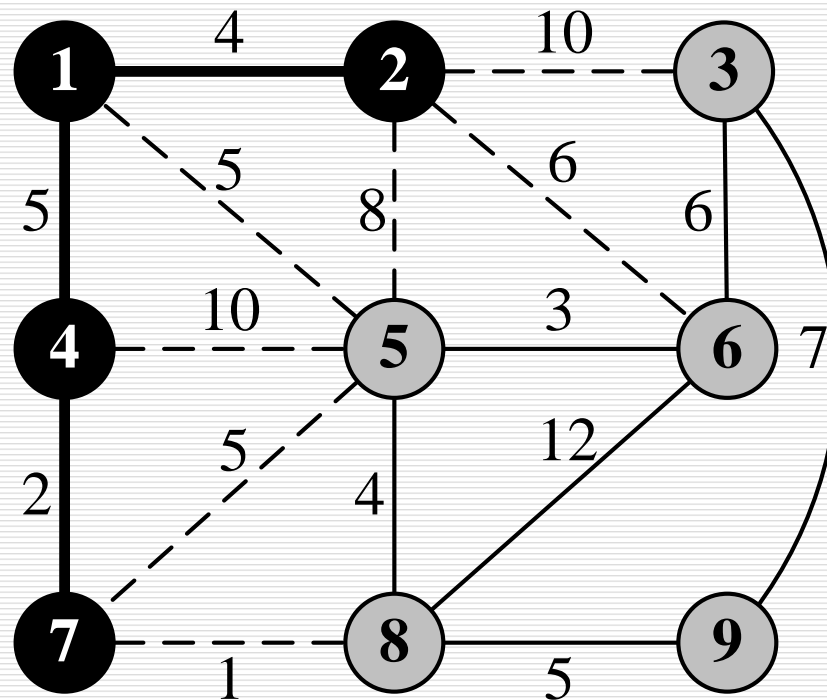
---





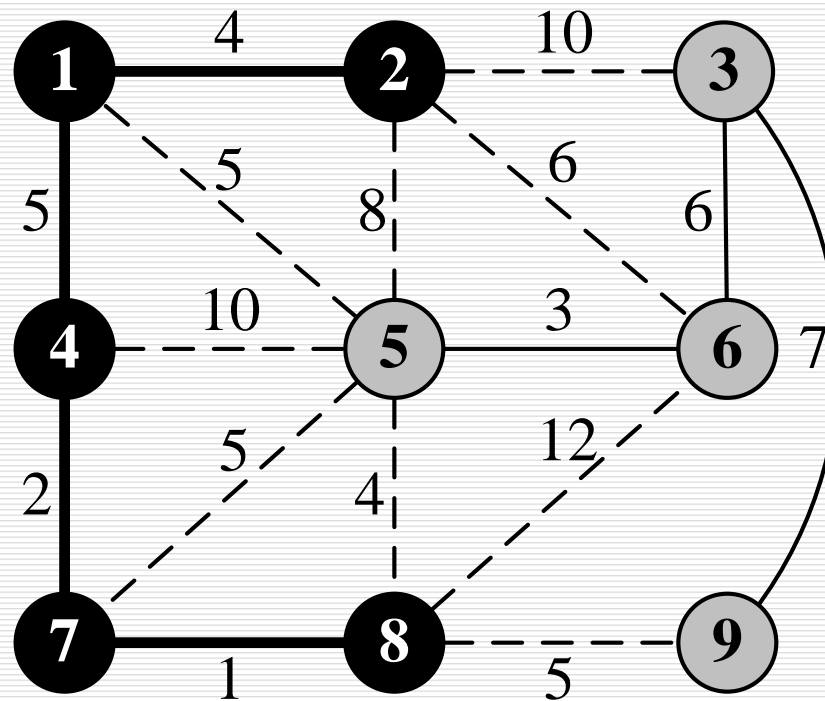
# Αλγόριθμος Prim: παράδειγμα

---



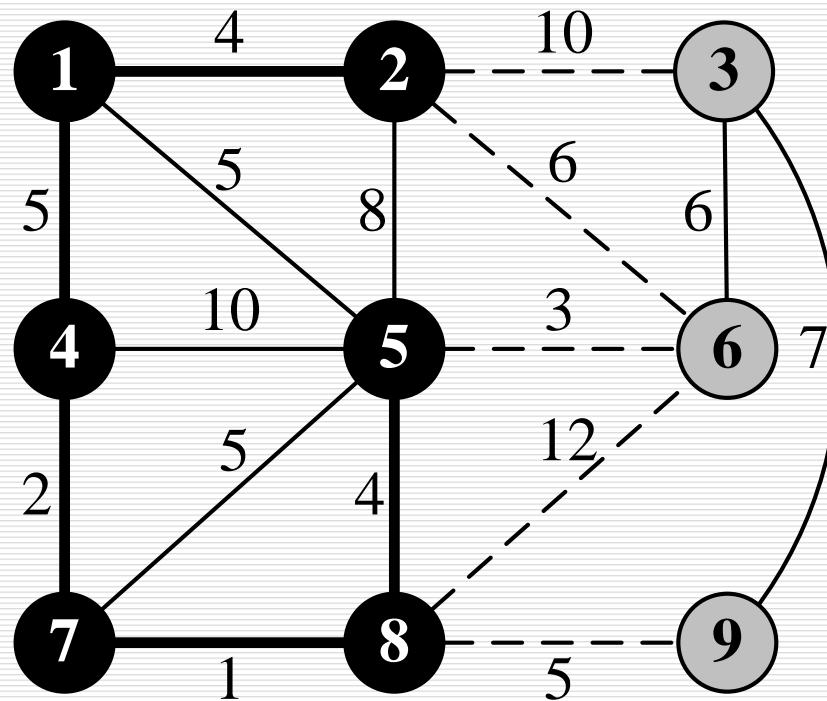
# Αλγόριθμος Prim: παράδειγμα

---



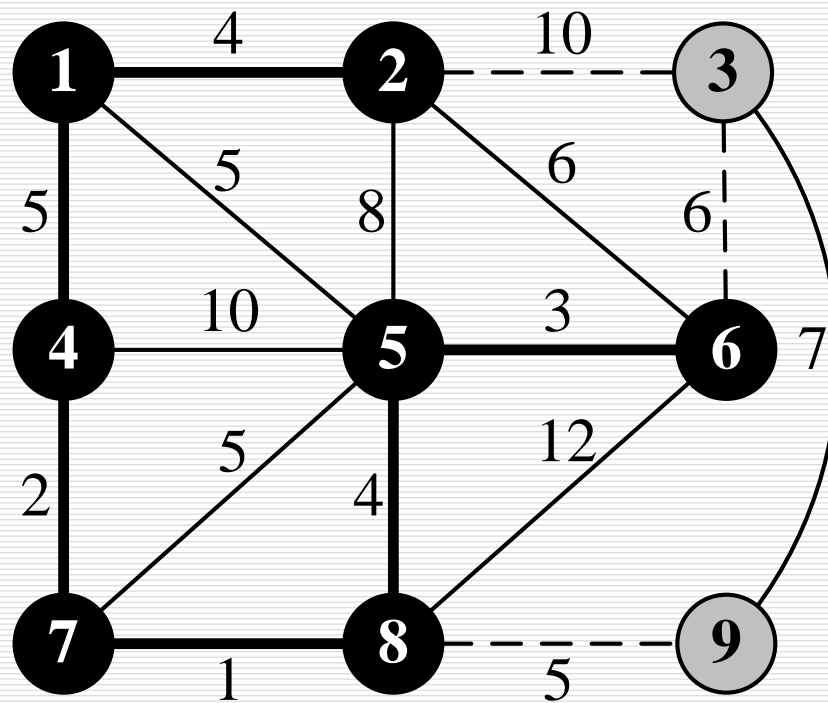
# Αλγόριθμος Prim: παράδειγμα

---



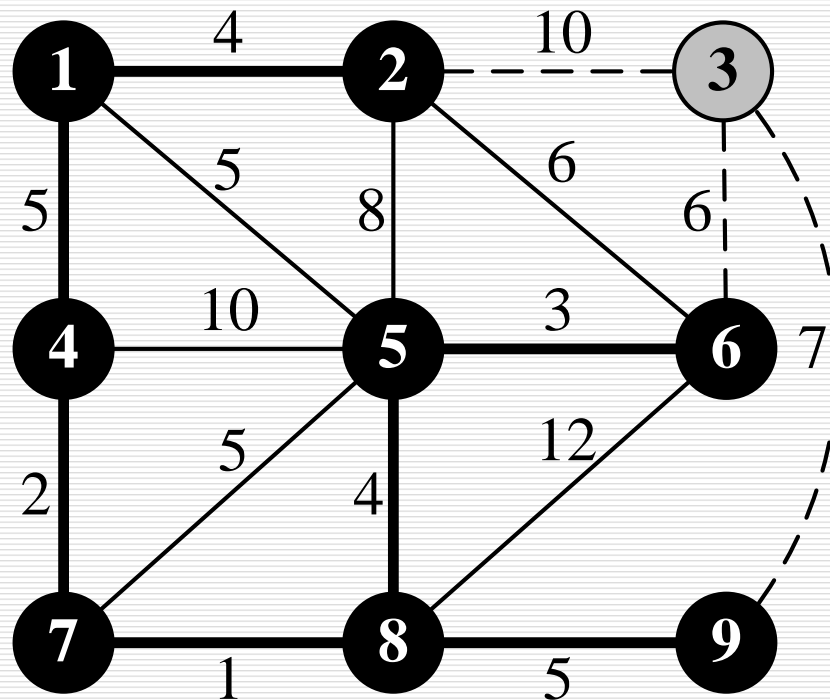
# Αλγόριθμος Prim: παράδειγμα

---



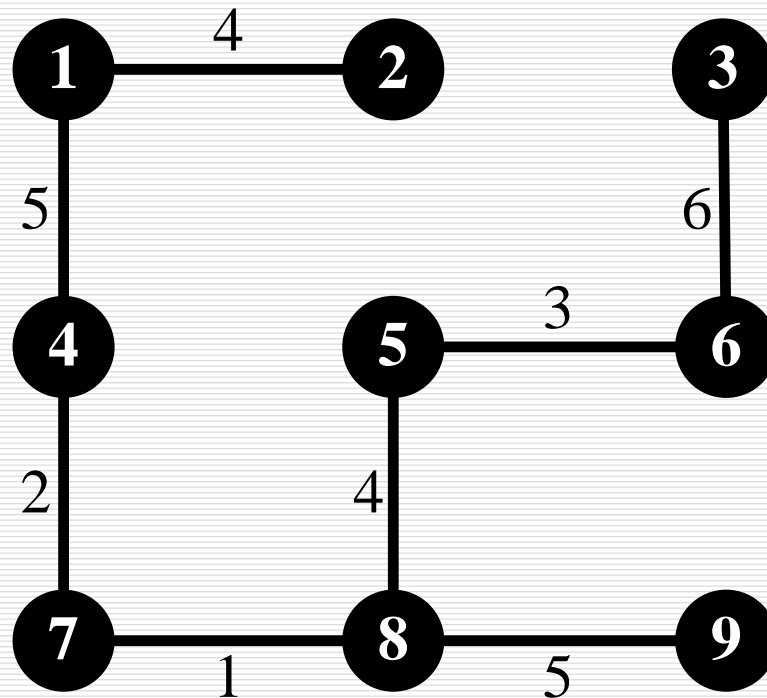
# Αλγόριθμος Prim: παράδειγμα

---



# Αλγόριθμος Prim: παράδειγμα

---



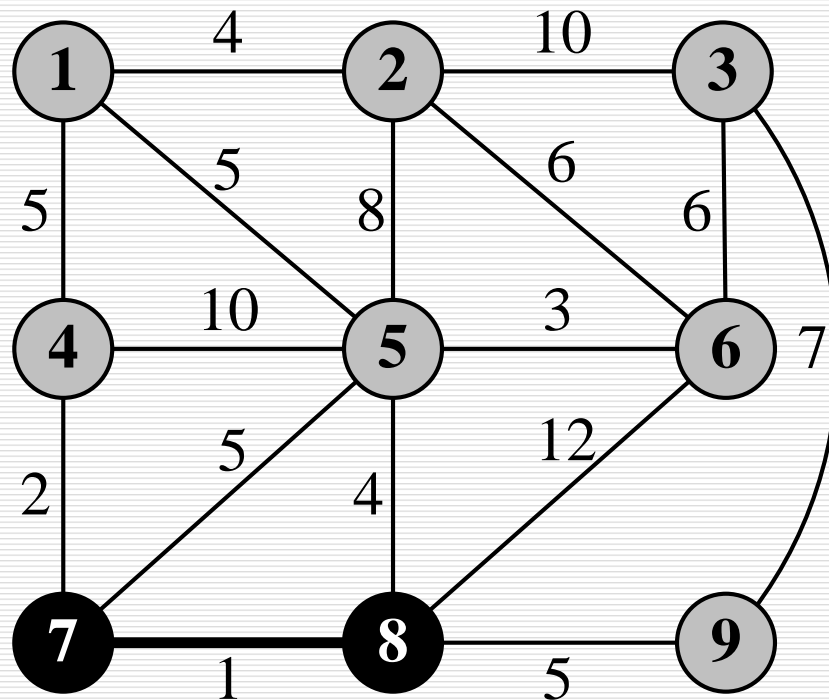
# Αλγόριθμος Prim

---

- Επιλέγεται αρχικός κόμβος, έστω  $s$ . Αρχικοποίηση:  
 $\text{dist}(s) := 0$ ; **for each**  $v \neq s$  **do**  $\text{dist}(v) := \infty$
- Κάθε φορά επιλέγεται ο κόμβος, έστω  $w$ , με την **ελάχιστη απόσταση από το μέχρι στιγμής κατασκευασμένο δένδρο**, και προστίθεται στο δένδρο. Ενημερώνονται οι αποστάσεις των υπόλοιπων κόμβων από το δένδρο με βάση το κόστος των ακμών  $(w, u_i)$ :  
  
 $\text{if } \text{cost}(w, u_i) < \text{dist}(u_i) \text{ then}$   
 $\quad \text{dist}(u_i) := \text{cost}(w, u_i)$
- Μεγάλη ομοιότητα με Dijkstra (πού διαφέρουν;)
- **Πολυπλοκότητα:  $O(|V|^2)$ ,  $O(|E|\log|V|)$ ,  $O(|E|+|V|\log|V|)$**

# Αλγόριθμος Kruskal: παράδειγμα

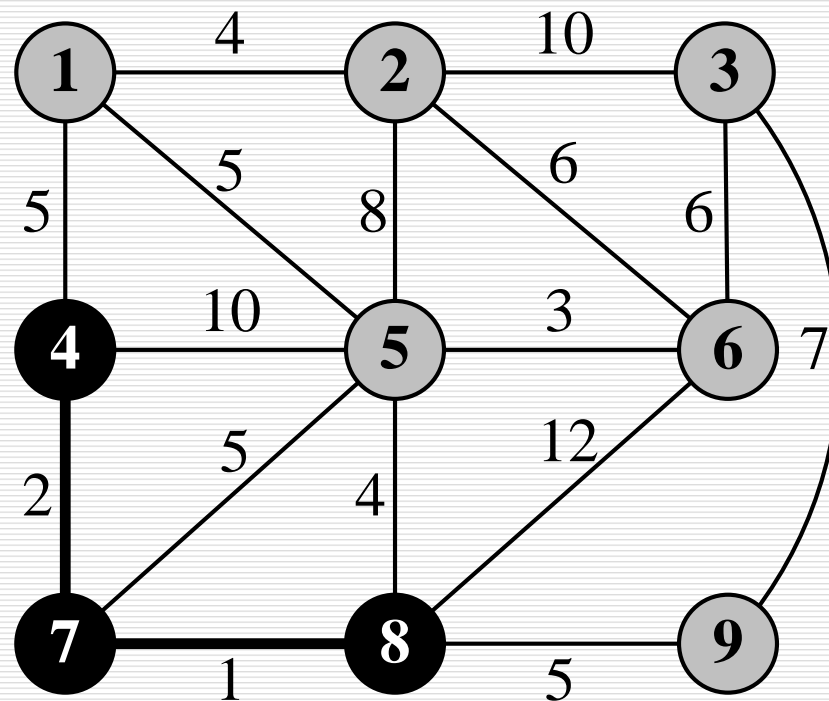
---





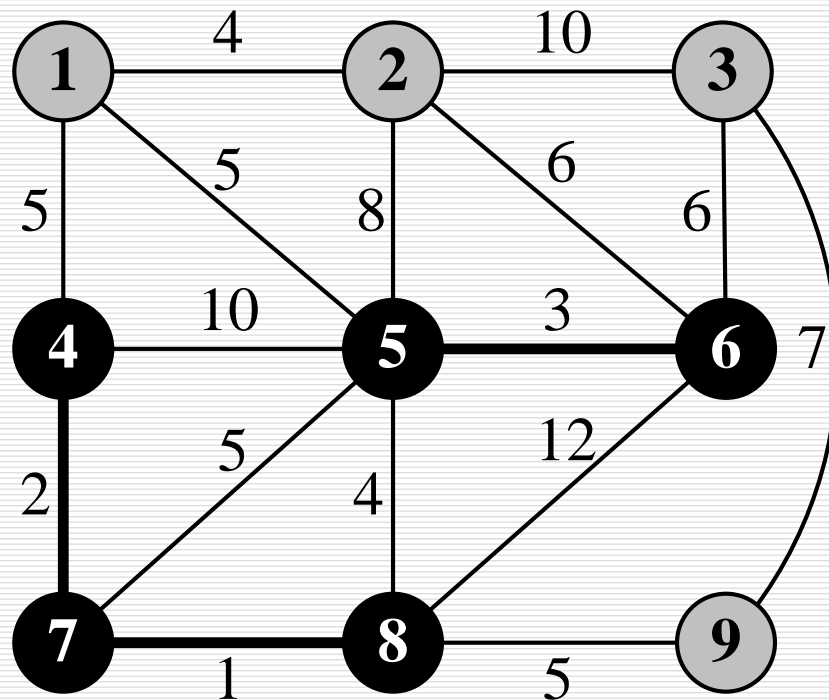
# Αλγόριθμος Kruskal: παράδειγμα

---



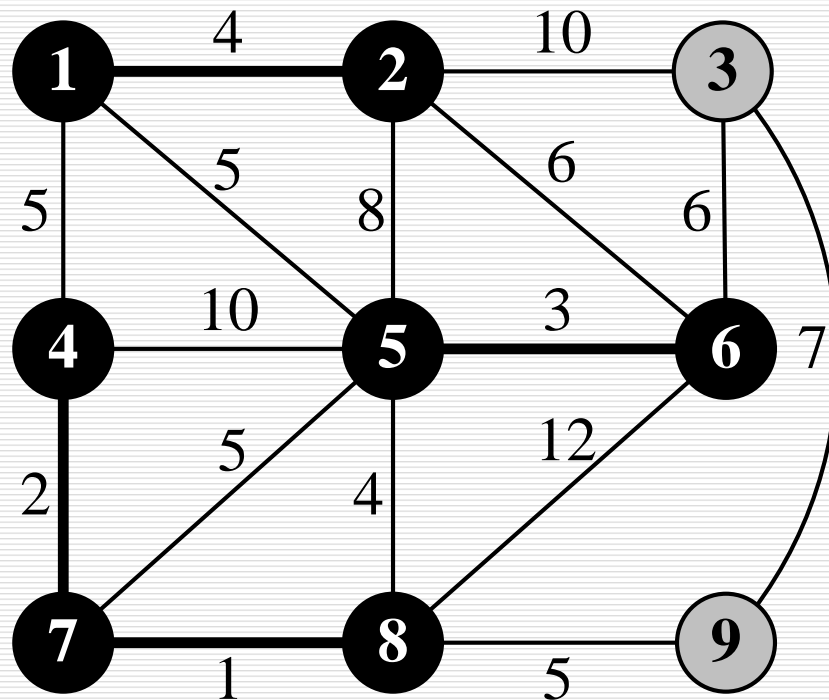
# Αλγόριθμος Kruskal: παράδειγμα

---



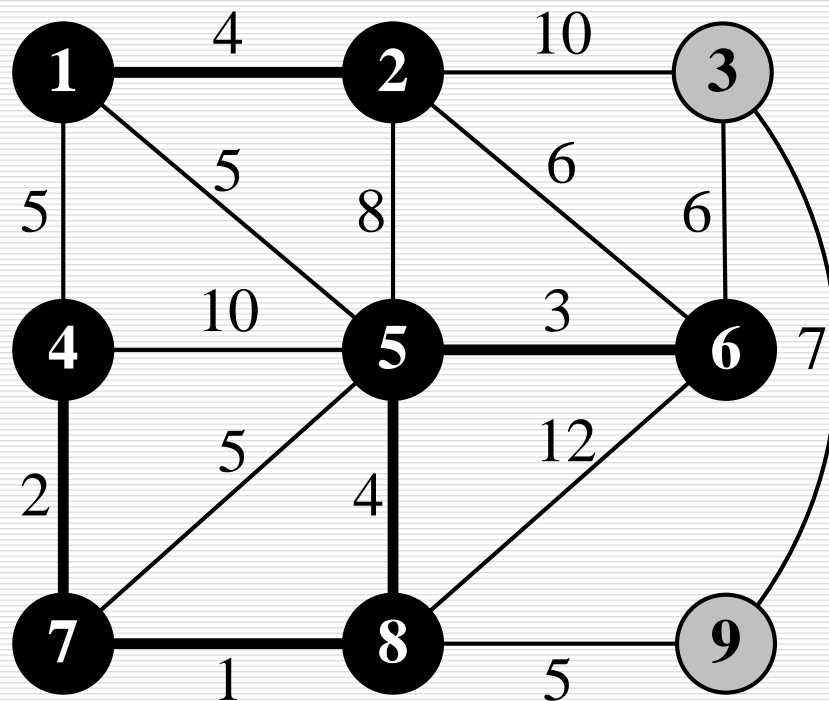
# Αλγόριθμος Kruskal: παράδειγμα

---



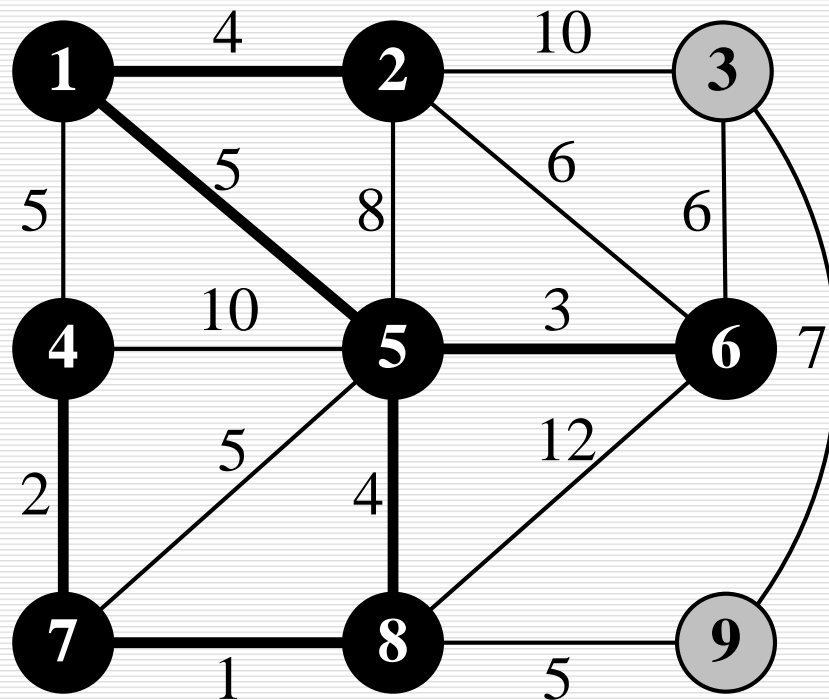
# Αλγόριθμος Kruskal: παράδειγμα

---



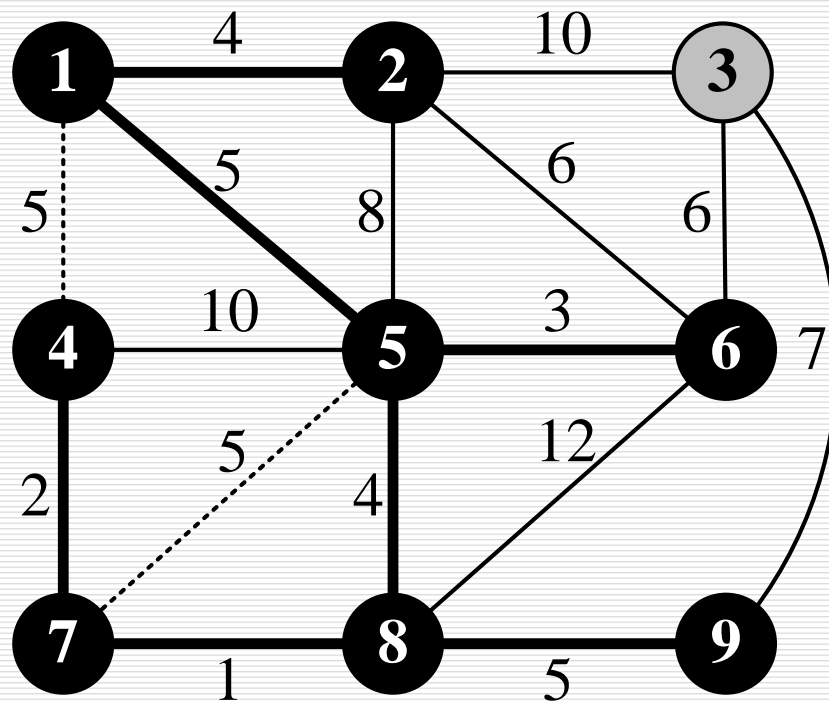
# Αλγόριθμος Kruskal: παράδειγμα

---



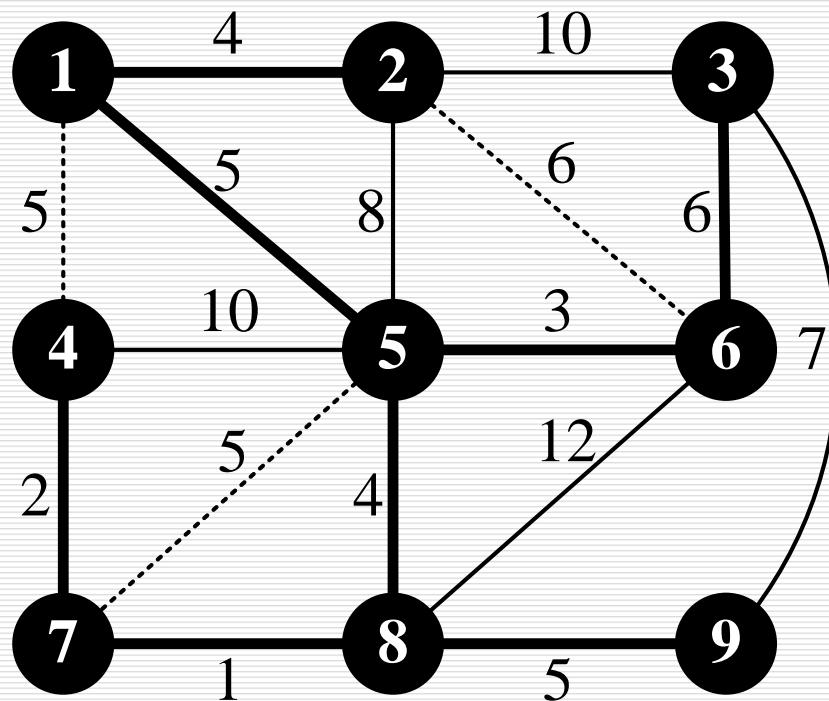
# Αλγόριθμος Kruskal: παράδειγμα

---



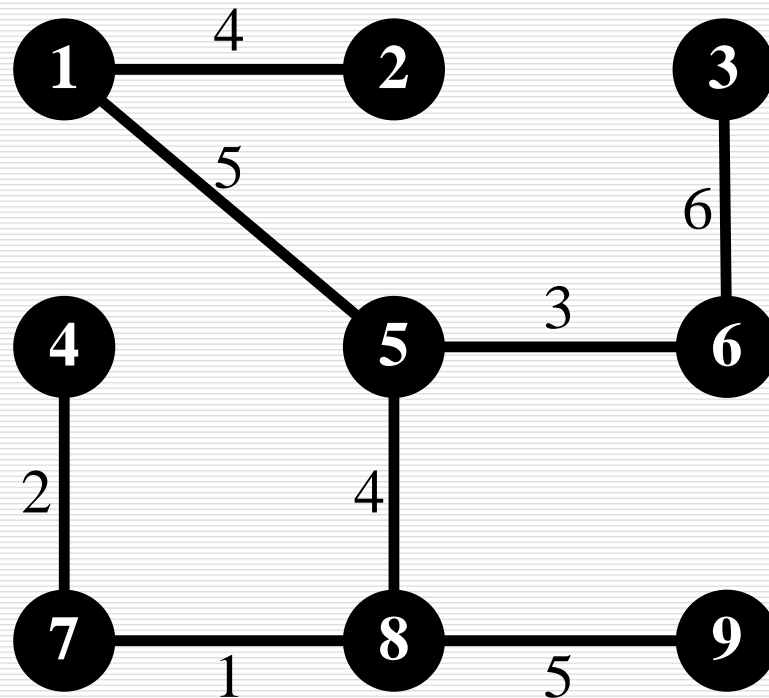
# Αλγόριθμος Kruskal: παράδειγμα

---



# Αλγόριθμος Kruskal: παράδειγμα

---





# Αλγόριθμος Kruskal

---

- Οι ακμές ταξινομούνται σε αύξουσα σειρά κόστους. Κάθε φορά επιλέγεται η ακμή ελαχίστου κόστους και αν δε δημιουργεί κύκλο στο μέχρι στιγμής δάσος προστίθεται σε αυτό, αλλιώς απορρίπτεται.
- Για αποδοτική υλοποίηση, η ύπαρξη κύκλου ελέγχεται με χρήση πράξεων συνόλων (UNION-FIND: αρκεί χρήση Union by Size/Rank).
- **Πολυπλοκότητα:  $O(|E|\log|V|)$**  (γιατί;)

# Κοινή ιδέα Prim-Kruskal

---

Έστω ο αρχικός γράφος  $G=(V, E)$ .

Ξεκινώντας από τον γράφο  $G'=(V, \emptyset)$  που περιέχει όλους τους κόμβους του  $G$  αλλά καθόλου ακμές, και ενώνοντας επαναληπτικά δύο **οποιαδήποτε** συμπληρωματικά υποσύνολα κόμβων  $S$  και  $V-S$  που ακόμη δεν έχουν ακμή μεταξύ τους με την **ελαφρύτερη δυνατή ακμή** από το  $E$ , καταλήγουμε σε ελάχιστο συνδετικό δένδρο.

# Γιατί δουλεύει η ιδέα;

---

## Θεώρημα.

Ένα σύνολο ακμών  $A$  που είναι *υποσχόμενο* (δηλ. υποσύνολο ενός MST) παραμένει υποσχόμενο αν του προσθέσουμε την **ελαφρύτερη ακμή  $e=(u,v)$**  που συνδέει μια συνεκτική συνιστώσα (connected component)  $V_i$  του τρέχοντος υπογράφου (που ορίζεται από τους κόμβους του  $V$  και τις ακμές του  $A$ ) με τον υπόλοιπο υπογράφο  $V-V_i$ .

*Απόδειξη.* Θεωρούμε ένα MST  $T$  που είναι υπερσύνολο του  $A$ . Έστω ότι η  $e$  δεν ανήκει στο  $T$ , τότε υπάρχει μονοπάτι  $p$  που συνδέει  $u,v$  στο  $T$ . Έστω  $e'$  στο  $p$  που διασχίζει τομή  $(V_i, V-V_i)$ . Ισχύει  **$\text{cost}(e) \leq \text{cost}(e')$** , επομένως:

**ανταλλαγή  $e, e' \Rightarrow$  MST  $T'$  που περιέχει την  $e$**

# Bonus: αλγόριθμος Boruvka

---

- Λειτουργεί σε γύρους. Αρχικά κάθε κόμβος είναι συνιστώσα μόνος του.
- Σε κάθε γύρο, *κάθε* συνεκτική συνιστώσα συνδέεται με την **ελαφρύτερη** δυνατή ακμή με κάποια από τις υπόλοιπες συνιστώσες. Χρειάζεται τρόπος επίλυσης 'ισοπαλιών'.

**Πολυπλοκότητα:**  $O(|E| \log|V|)$  (σε κάθε γύρο το πλήθος συνιστωσών μειώνεται στο μισό).

Προσφέρεται για **παράλληλη / κατανεμημένη** υλοποίηση.

# Δύο σημαντικές τεχνικές

---

- **Άπληστοι (greedy) αλγόριθμοι**: «χτίσιμο» λύσης σταδιακά, από μικρότερα προς μεγαλύτερα υποπροβλήματα. Σε κάθε στάδιο **αμετάκλητη** επιλογή, δίνει βέλτιστη λύση για αντίστοιχο υποπρόβλημα.
  - **Dijkstra, Prim, Kruskal, Boruvka**
- **Δυναμικός προγραμματισμός**: «χτίσιμο» λύσης σταδιακά, συνδυάζοντας βέλτιστες λύσεις μικρότερων υποπροβλημάτων ώστε να προκύψει βέλτιστη λύση μεγαλύτερων (**αρχή βελτιστότητας υπο-λύσεων**).
  - **Bellman-Ford**
- Σύγκριση με «**διαίρει και κυρίευε**»: στη ΔκΚ τα υποπροβλήματα είναι ανεξάρτητα, στον ΔΠ και στους άπληστους τα υποπροβλήματα έχουν επικάλυψη.

# Προβλήματα Γράφων στην Κλάση **P**

---

- Κύκλος Euler
- Προσβασιμότητα (reachability) + Διάσχιση (traversal): DFS, BFS, ...
- Συνεκτικές συνιστώσες (connected components)
- Συντομότερα μονοπάτια (shortest paths)
- Ελάχιστο συνδετικό δένδρο (minimum spanning tree)
- Μέγιστη ροή (maximum flow) [7<sup>ο</sup> εξ.]
- Τέλειο ταίριασμα (perfect matching) [7<sup>ο</sup> εξ.]

# NP-πλήρη Προβλήματα Γράφων

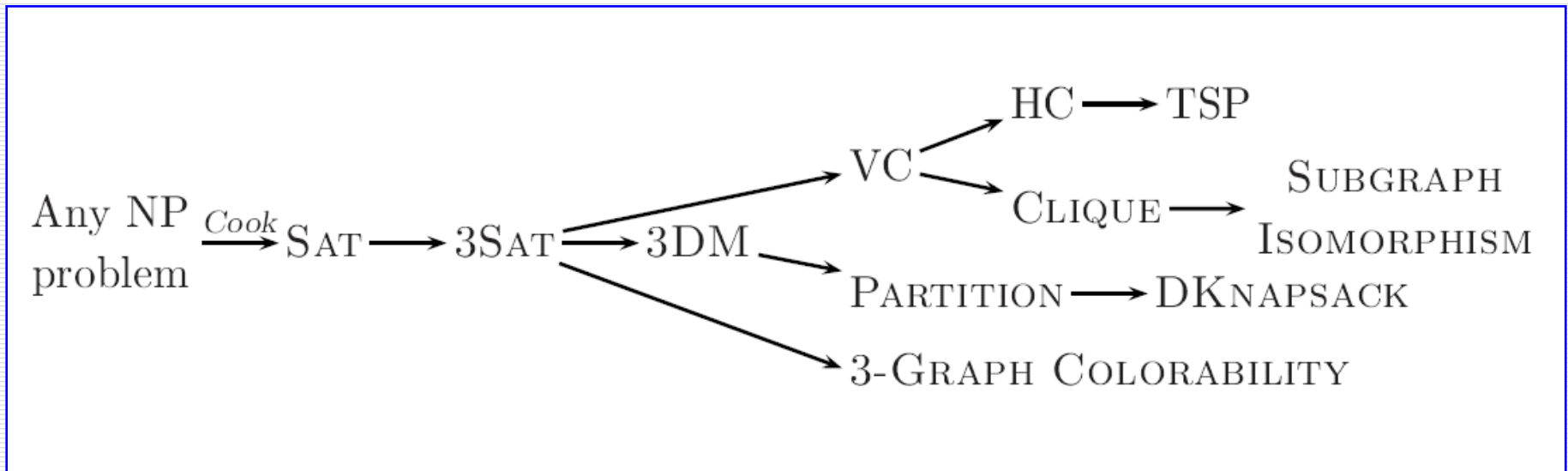
---

- VERTEX COVER (VC)
- CLIQUE
- HAMILTON CIRCUIT (HC)
- TRAVELING SALESMAN (TSP)
- 3-COLORABILITY
- SUBGRAPH ISOMORPHISM
- 3-DIMENSIONAL MATCHING (3DM)

# NP-πλήρη Προβλήματα Γράφων

---

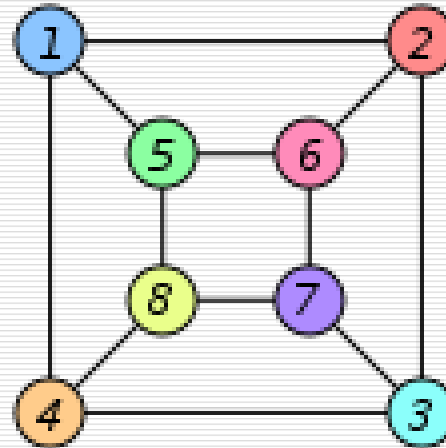
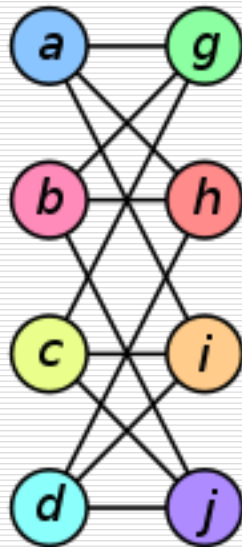
Απόδειξη **NP**-πληρότητας: *αναγωγές*





# «Ενδιάμεση» Πολυπλοκότητα;

---



Ισομορφισμός γράφων: δεν είναι NP-πλήρες πρόβλημα (κάτω από γενικά παραδεκτές υποθέσεις)

# Συμπεράσματα

---

- Αρκετά προβλήματα γράφων λύνονται γρήγορα: διάσχιση (προσβασιμότητα), συνεκτικές συνιστώσες, ελάχιστες διαδρομές, ελάχιστο συνδετικό δένδρο, κύκλος Euler, τέλειο ταίριασμα, μέγιστη ροή, ...
- Πολλά προβλήματα φαίνεται να μην λύνονται γρήγορα: VERTEX COVER, CLIQUE, HAMILTON CIRCUIT, TRAVELING SALESMAN, 3-COLORABILITY, SUBGRAPH ISOMORPHISM, 3-DIMENSIONAL MATCHING, ...
- Κάποια από αυτά λύνονται γρήγορα σε ειδικές περιπτώσεις, ή προσεγγιστικά. Εντατική έρευνα, πολλά ανοιχτά ερωτήματα.