ΜΕΘΟΔΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΤΗ ΔΙΟΙΚΗΤΙΚΗ ΕΠΙΣΤΗΜΗ (ΜΕ.ΒΕ.Δ.Ε)

Emmanouil Zachariadis, Assistant Professor
Department of Management Science and Technology,
Athens University of Economics and Business
E: ezach@aueb.gr
A: Evelpidon 47A and Lefkados 33 street, 8th floor, Room 811, 11362, Athens, Greece
T: +30-210-8203 467

3η ΣΥΝΔΥΑΣΤΙΚΗ ΔΟΜΗ:
ΟΜΑΔΟΠΟΙΗΣΗ ΣΤΟΙΧΕΙΩΝ → ΠΡΟΒΛΗΜΑΤΑ ΕΠΙΛΟΓΗΣ

$3^{η}$ ΣΥΝΔΥΑΣΤΙΚΗ ΔΟΜΗ:
ΕΠΙΛΟΓΗ ΣΤΟΙΧΕΙΩΝ → ΠΡΟΒΛΗΜΑΤΑ ΕΠΙΛΟΓΗΣ

Προβλήματα Επιλογής: Δεδομένου ενός συνόλου στοιχείων, καλείσαι να επιλέξεις κάποια από τα στοιχεία με σκοπό να βελτιστοποιήσεις κάποιο αντικειμενικό στόχο. Λόγω της ύπαρξης περιορισμών δεν είναι δυνατό να επιλεγούν όλα τα στοιχεία του συνόλου.

ΤΟ ΠΡΟΒΛΗΜΑ SET COVERING

- Περιγραφή του Προβλήματος Set Covering:
  - <u>Δεδομένα</u>:

    Σύνολο U (Σύμπαν) που αποτελείται από m στοιχεία, $U = \{1,2,3, \ldots, m\}$

    Σύνολο $S_{comp} = \{S_1, S_2, \ldots, S_n\}$, με $S_i \subset U$

    $\exists S \in S_{comp}, : u \in S, \forall u \in U,$

    δηλαδή για κάθε $u \in U$, υπάρχει τουλάχιστον ένα στοιχείο (σύνολο S) που ανήκει στο σύνολο S_{comp} και περιλαμβάνει το στοιχείο u
  - <u>Στόχος</u>: Να βρεθεί ο *μικρότερος αριθμός υποσυνόλων* $S_i \in S_{comp}$, τα οποία θα περιέχουν όλα τα στοιχεία του συνόλου U

ΤΟ ΠΡΟΒΛΗΜΑ SET COVERING

- Αντικειμενική Συνάρτηση:

$$\min \sum_{S \in S_{comp}} x_S$$

- Περιορισμοί:

$$\sum_{S:u \in S} x_S \geq 1, \forall u \in U$$

Για κάθε στοιχείο του U, βρες όλα τα υποσύνολα S, τα οποία το περιέχουν (S:u∈S). Τουλάχιστον ένα από αυτά πρέπει να επιλεγεί ($\sum_{S:u\in S} x_S \geq 1$).

$$x_S \in \{0, 1\}, \forall S \in S_{comp}$$

Ένα υποσύνολο $\forall S \in S_{comp}$ είτε επιλέγεται είτε όχι

ΕΦΑΡΜΟΓΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ SET COVERING: ΥΠΗΡΕΣΙΕΣ ΥΓΕΙΑΣ (Health Care Services)

- Υπάρχουν 10 δήμοι στην ευρύτερη περιοχή της Αθήνας που έχουν ζητήσει να υπάρχει ένα σταθμευμένο ασθενοφόρο σε κάποιο σημείο της περιοχής τους ώστε να είναι έτοιμο να επέμβει σε περίπτωση που συμβεί κάποιο έκτακτο περιστατικό.

- Το Υπουργείο Υγείας απάντησε ότι το ζήτημα δεν είναι να υπάρχει εγκατεστημένο ένα ασθενοφόρο σε κάθε δήμο αλλά να μπορεί ένα ασθενοφόρο (που μπορεί να είναι εγκατεστημένο ακόμα και σε άλλο δήμο) να εξυπηρετήσει έγκαιρα τη ζήτηση που θα προκύψει.

- Ο βαθμός ποιότητας του οδικού δικτύου που συνδέει τους δήμους καθώς και ο πληθυσμός που διαμένει σε κάθε ένα από αυτούς καθορίζει αν ένα ασθενοφόρο που σταθμεύει σε ένα συγκεκριμένο δήμο μπορεί να προσφέρει τις υπηρεσίες του και σε άλλους δήμους.

ΕΦΑΡΜΟΓΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ SET COVERING: ΥΠΗΡΕΣΙΕΣ ΥΓΕΙΑΣ (Health Care Services)

- Ένα ασθενοφόρο του δήμου 1 μπορεί να εξυπηρετήσει **και** το δήμο 2
- Ένα ασθενοφόρο του δήμου 2 μπορεί να εξυπηρετήσει **και** τους δήμους 1,3
- Ένα ασθενοφόρο του δήμου 3 μπορεί να εξυπηρετήσει **και** τους δήμους 4,5,6
- Ένα ασθενοφόρο του δήμου 4 μπορεί να εξυπηρετήσει **και** τους δήμους 2,3,9
- Ένα ασθενοφόρο του δήμου 5 μπορεί να εξυπηρετήσει **και** τους δήμους 4,6,8
- Ένα ασθενοφόρο του δήμου 6 μπορεί να εξυπηρετήσει **και** το δήμο 7
- Ένα ασθενοφόρο του δήμου 7 μπορεί να εξυπηρετήσει **και** το δήμο 8
- Ένα ασθενοφόρο του δήμου 8 μπορεί να εξυπηρετήσει **και** τους δήμους 5,7,10
- Ένα ασθενοφόρο του δήμου 9 μπορεί να εξυπηρετήσει **και** τους δήμους 4,8
- Ένα ασθενοφόρο του δήμου 10 μπορεί να εξυπηρετήσει **και** το δήμο 8

ΕΦΑΡΜΟΓΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ SET COVERING: ΥΠΗΡΕΣΙΕΣ ΥΓΕΙΑΣ (Health Care Services)

- Στόχος του Υπουργείου Υγείας είναι να βρεθεί ο ελάχιστος αριθμός ασθενοφόρων που πρέπει να σταθμεύσουν σε συγκεκριμένους δήμους ώστε να καλύπτεται η ζήτηση όλων των δήμων που έχουν κάνει αίτημα.

- Να σχεδιαστεί μια ΠΣΕ για την επίλυση του συγκεκριμένου προβλήματος.

ΕΦΑΡΜΟΓΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ SET COVERING: ΥΠΗΡΕΣΙΕΣ ΥΓΕΙΑΣ (Health Care Services)

- <u>Σύμπαν</u>: Το σύνολο των 10 δήμων, $U = \{1,2 , \dots 10\}$
- Σύνολο S_{comp}: Ένα σύνολο κάθε στοιχείο του οποίου αντιπροσωπεύει ένα δήμο και αποτελείται από ένα υποσύνολο δήμων που μπορεί να εξυπηρετήσει
- Επομένως:

$$S_{comp} = \{S_1, S_2, \dots , S_{10}\}$$

- όπου
  - $S_1 : \{1,2\}$
  - $S_2 : \{1,2,3\}$
  - $S_3 : \{3,4,5,6\}$
  - $S_4 : \{2,3,4,9\}$
  - $S_5 : \{4,5,6,8\}$
  - $S_6 : \{6,7\}$
  - $S_7 : \{7,8\}$
  - $S_8 : \{5,7,8,10\}$
  - $S_9 : \{4,8,9\}$
  - $S_{10} : \{8,10\}$

ΕΦΑΡΜΟΓΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ SET COVERING: ΥΠΗΡΕΣΙΕΣ ΥΓΕΙΑΣ (Health Care Services)

- Αντικειμενική Συνάρτηση
  $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$

- Περιορισμοί:

$x1 + x2 \geq 1$ (Εξυπηρέτηση Δήμου 1)
$x1 + x2 + x4 \geq 1$ (Εξυπηρέτηση Δήμου 2)
$x2 + x3 + x4 \geq 1$ (Εξυπηρέτηση Δήμου 3)
$x3 + x4 + x5 + x9 \geq 1$ (Εξυπηρέτηση Δήμου 4)
$x3 + x5 + x8 \geq 1$ (Εξυπηρέτηση Δήμου 5)
$x3 + x5 + x6 \geq 1$ (Εξυπηρέτηση Δήμου 6)
$x6 + x7 + x8 \geq 1$ (Εξυπηρέτηση Δήμου 7)
$x5 + x7 + x8 + x9 + x10 \geq 1$ (Εξυπηρέτηση Δήμου 8)
$x4 + x9 \geq 1$ (Εξυπηρέτηση Δήμου 9)
$x8 + x10 \geq 1$ (Εξυπηρέτηση Δήμου 10)

$S1$: {1,2}
$S2$: {1,2,3}
$S3$: {3,4,5,6}
$S4$: {2,3,4,9}
$S5$: {4,5,6,8}
$S6$: {6,7}
$S7$: {7,8}
$S8$: {5,7,8,10}
$S9$: {4,8,9}
$S10$: {8,10}

ΕΦΑΡΜΟΓΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ SET COVERING: ΥΠΗΡΕΣΙΕΣ ΥΓΕΙΑΣ (Health Care Services)

- **Μορφή Λύσης**: Ένα υποσύνολο των στοιχείων του S_{comp}
- **Στοιχείο Λύσης**: Το στοιχείο λύσης που προστίθεται στην μερική λύση είναι ένα υποσύνολο $S \in S_{comp}$, το οποίο περιλαμβάνει τους δήμους που εξυπηρετούνται από ένα ασθενοφόρο
- **Κριτήριο Επιλογής**: Επίλεξε, σε κάθε επανάληψη, το υποσύνολο S, το οποίο μεγιστοποιεί την κάλυψη (εξυπηρέτηση) των δήμων που δεν εξυπηρετούνται.
- **Αντικειμενική Συνάρτηση**: Ο συνολικός αριθμός των υποσυνόλων του S που θα επιλεγούν

ΕΦΑΡΜΟΓΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ SET COVERING: ΥΠΗΡΕΣΙΕΣ ΥΓΕΙΑΣ (Health Care Services)

- **Επανάληψη 1**: Υπάρχουν 4 σύνολα (S_3,S_4,S_5,S_8) που ικανοποιούν το «κριτήριο επιλογής». Επιλέγεται στοχαστικά το σύνολο S_3, που σημαίνει ότι το πρώτο ασθενοφόρο θα εγκατασταθεί στο δήμο 3 και θα μπορεί να εξυπηρετήσει περιστατικά των δήμων 3,4,5,6. Άρα η ημιτελής λύση διαμορφώνεται ως $\boldsymbol{s} = \{\boldsymbol{S_3}\}$.

- **Επανάληψη 2**: Το σύνολο $\boldsymbol{S_8}$ ικανοποιεί το «κριτήριο επιλογής». Άρα η λύση τροποπειείται ως εξής $\boldsymbol{s} = \{\boldsymbol{S_3}, \boldsymbol{S_8}\}$.

ΕΦΑΡΜΟΓΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ SET COVERING: ΥΠΗΡΕΣΙΕΣ ΥΓΕΙΑΣ (Health Care Services)

- **Επανάληψη 3**: Υπάρχουν 3 σύνολα ($\boldsymbol{S_1}, \boldsymbol{S_2}, \boldsymbol{S_4}$) που ικανοποιούν το «κριτήριο επιλογής». Επιλέγεται στοχαστικά το σύνολο $\boldsymbol{S_1}$. Επομένως, $\boldsymbol{s} = \{\boldsymbol{S_3}, \boldsymbol{S_8}, \boldsymbol{S_1}\}$.

- **Επανάληψη 4**: Υπάρχουν 2 σύνολα ($\boldsymbol{S_4}, \boldsymbol{S_9}$) που ικανοποιούν το «κριτήριο επιλογής». Επιλέγεται στοχαστικά το σύνολο $\boldsymbol{S_4}$. Επομένως, η τελική λύση είναι η $\boldsymbol{s} = \{\boldsymbol{S_3}, \boldsymbol{S_8}, \boldsymbol{S_1}, \boldsymbol{S_4}\}$.

Άρα η λύση μας είναι η $\boldsymbol{s} = \{\boldsymbol{S_3}, \boldsymbol{S_8}, \boldsymbol{S_1}, \boldsymbol{S_4}\}$ και το κόστος της λύσης μας είναι ίσο με 4, δηλαδή θα εγκατασταθούν 4 ασθενοφόρα στους δήμους 3,9,2 και 7 ώστε να υπάρχει η δυνατότητα να εξυπηρετούνται και οι 10 δήμοι που έχουν κάνει αίτημα.

ΕΠΙΛΟΓΗ ΠΡΟΣΩΠΙΚΟΥ

- Η Διοίκηση ενός οργανισμού έχει αποφασίσει την πρόσληψη υπαλλήλων με σκοπό να δουλέψουν σε ένα συγκεκριμένο project. Η πετυχημένη ολοκλήρωση του συγκεκριμένου project απαιτεί τις ακόλουθες δεξιότητες από το σύνολο (όχι από τον/την καθένα) των υπαλλήλων που θα προσληφθούν:
    1. Επικοινωνία
    2. Γνώσεις Επιχειρησιακής Έρευνας
    3. IT γνώσεις
    4. Εξόρυξη Δεδομένων
    5. Γνώσεις Management
    6. Ηγετικές Ικανότητες
    7. Δημιουργική Σκέψη
    8. Ικανότητες Σχεδιασμού και Οργάνωσης
    9. Ομαδική Δουλειά
    10. Προσωπικές Αξίες

ΕΠΙΛΟΓΗ ΠΡΟΣΩΠΙΚΟΥ

- Υπάρχουν 10 υποψήφιοι, καθένας από τους οποίους χαρακτηρίζεται από τα ακόλουθες δεξιότητες: Ο **Υπάλληλος 1** χαρακτηρίζεται από τις δεξιότητες 1 και 2, ο **Υπάλληλος 2** από τις δεξιότητες 1,2 και 3, ο **Υπάλληλος 3** από τις δεξιότητες 3,4,5 και 6, ο **Υπάλληλος 4** από τις δεξιότητες 2,3,4 και 9, ο **Υπάλληλος 5** από τις δεξιότητες 4,5,6 και 8, ο **Υπάλληλος 6** από τις δεξιότητες 6 και 7, ο **Υπάλληλος 7** από τις δεξιότητες 7 και 8, ο **Υπάλληλος 8** από τις δεξιότητες 5,7,8 και 10, ο **Υπάλληλος 9** από τις δεξιότητες 4,8,9 και ο **Υπάλληλος 10** από τις δεξιότητες 8 και 10.

- Η Διοίκηση επιθυμεί από έναν διοικητικό επιστήμονα να σχεδιάσει μια ΠΣΕ για το βέλτιστο καθορισμό της ομάδας έργου για το συγκεκριμένο project, στοχεύοντας στην πρόσληψη του ελάχιστου αριθμού υπαλλήλων έτσι ώστε το σύνολο των υπαλλήλων που θα προσληφθούν να έχουν τις δεξιότητες που απαιτούνται για την επιτυχημένη ολοκλήρωση του project.

ΕΠΙΛΟΓΗ ΠΡΟΣΩΠΙΚΟΥ

- Μορφή Λύσης: Αριθμός ομαδοποιήσεων (υποσυνόλων S_j) των 10 δεξιοτήτων.
- Στοιχείο Λύσης: Μία ομαδοποίηση S_j δεξιοτήτων ενός υπαλλήλου j.
- Κριτήριο Επιλογής: Επίλεξε, σε κάθε επανάληψη, μια ομαδοποίηση S_j που καλύπτει με τον καλύτερο τρόπο τις δεξιότητες που δεν έχουν «καλυφθεί» από τους υπαλλήλους που έχουν ήδη προσληφθεί, μέχρι να «καλυφθούν» και οι 10 δεξιότητες που απαιτούνται για την πραγματοποίηση του project.
- Κριτήριο Αξιολόγησης: Ο συνολικός αριθμός των υπαλλήλων που θα επιλεγούν για να εργαστούν στο συγκεκριμένο project.

Ίδια αναπαράσταση ως set covering πρόβλημα με την εφαρμογή για τα ασθενοφόρα, ίδια δεδομένα, ίδια ΠΣΕ και όπως είναι λογικό το ίδιο αποτέλεσμα.

ΕΠΙΛΟΓΗ ΠΡΟΣΩΠΙΚΟΥ

- Πώς θα άλλαζε το πρόβλημα αν οι Υπάλληλοι είχαν διαφορετικό μισθό και ως αντικειμενική συνάρτηση επιλέγαμε την ελαχιστοποίηση των συνολικών εξόδων της εταιρείας με σκοπό τη δημιουργία μίας ολοκληρωμένης ομάδας για το έργο;

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Αντικειμενική Συνάρτηση:

$$\min \sum_{S \in S_{comp}} w_S \cdot x_S$$

- Περιορισμοί:

$$\sum_{S:u \in S} x_S \geq 1, \forall u \in U$$

Για κάθε στοιχείο του U, βρες όλα τα υποσύνολα S, τα οποία το περιέχουν (S:u∈S). Τουλάχιστον ένα από αυτά πρέπει να επιλεγεί ($\sum_{S:u\in S} x_S \geq 1$).

$$x_S \in \{0, 1\}, \forall S \in S_{comp}$$

Ένα υποσύνολο $\forall S \in S_{comp}$ είτε επιλέγεται είτε όχι

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Σύμπαν: Δεξιότητες ($U = \{1,2,3,\dots,10\}$)
- Πιθανές προσλήψεις (Δυνητικοί Υπάλληλοι)

$$S_{comp} = \{S_1, S_2, \dots, S_{10}\}$$

- όπου
  - $S_1 : \{1,2\}$
  - $S_2 : \{1,2,3\}$
  - $S_3 : \{3,4,5,6\}$
  - $S_4 : \{2,3,4,9\}$
  - $S_5 : \{4,5,6,8\}$
  - $S_6 : \{6,7\}$
  - $S_7 : \{7,8\}$
  - $S_8 : \{5,7,8,10\}$
  - $S_9 : \{4,8,9\}$
  - $S_{10} : \{8,10\}$

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Αντικειμενική Συνάρτηση
$$1000 \cdot x_1 + 700 \cdot x_2 + 800 \cdot x_3 + 0 \cdot x_4 + 1150 \cdot x_5 + 780 \cdot x_6 + 800 \cdot x_7 + 1200 \cdot x_8 + 1500 \cdot x_9 + 750 \cdot x_9 + 1200 \cdot x_{10}$$

- Περιορισμοί:

$$\begin{gathered} x1 + x2 \geq 1 \\ x1 + x2 + x4 \geq 1 \\ x2 + x3 + x4 \geq 1 \\ x3 + x4 + x5 + x9 \geq 1 \\ x3 + x5 + x8 \geq 1 \\ x3 + x5 + x6 \geq 1 \\ x6 + x7 + x8 \geq 1 \\ x5 + x7 + x8 + x9 + x10 \geq 1 \\ x4 + x9 \geq 1 \\ x8 + x10 \geq 1 \end{gathered}$$

$S1$: {1,2}
$S2$: {1,2,3}
$S3$: {3,4,5,6}
$S4$: {2,3,4,9}
$S5$: {4,5,6,8}
$S6$: {6,7}
$S7$: {7,8}
$S8$: {5,7,8,10}
$S9$: {4,8,9}
$S10$: {8,10}

ΠΣΕ για το Weighted SET COVERING

- Βήμα 1:
  Αν το βάρος κάποιου υποψηφίου $S = S_{comp}$ στην αντικειμενική συνάρτηση είναι 0, τότε
  $$x_S = 1$$
  Διαγραφή όσων περιορισμών εμπλέκουν το x_S

  Σχόλιο: Η συγκεκριμένη επιλογή είναι δωρεάν. Αυτομάτως, όλοι οι περιορισμοί που εμπλέκουν τον επιλεγμένο υπάλληλο δεν έχουν λόγο ύπαρξης, οι συγκεκριμένες δεξιότητες είναι ήδη καλυμμένες από τον δωρεάν υπάλληλο.

ΠΣΕ για το Weighted SET COVERING

- Βήμα 2:
  Αν (α) το βάρος κάποιου υποψηφίου $S \in S_{comp}$ στην αντικειμενική συνάρτηση δεν είναι 0 και (β) δεν εμφανίζεται σε κανέναν από τους υπόλοιπους περιορισμούς, τότε

$$x_S = 0$$

  <u>Σχόλιο:</u> Θα έπρεπε να πληρώσουμε για κάποιον υπάλληλο, ο οποίος δεν μας καλύπτει οποιαδήποτε ακάλυπτη δεξιότητα. Δεν υπάρχει κανένας λόγος να τον προσλάβουμε

ΠΣΕ για το Weighted SET COVERING

- Βήμα 3:
  Επαναληπτικά και μέχρι να μην υπάρχουν άλλοι περιορισμοί:
    1. Για όλα τα $S \in S_{comp}$ που δεν έχουν λάβει ακόμα x_s, υπολόγισε:

    $$v_S = \frac{c_S}{d_S}$$

    Με d_S το πλήθος των περιορισμών που περιέχουν το σύνολο S

    2. Βρες τον υποψήφιο S^*με την ελάχιστη τιμή v_S

    3. Θέσε $x_{S^*} = 1$

    4. Διάγραψε όσους περιορισμούς εμπλέκουν τον S^*

    5. Βήμα 2 (Διαγραφή όσων υποψηφίων δεν είναι πλέον απαραίτητοι)

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Βήμα 1

$$1000 \cdot x_1 + 700 \cdot x_2 + 800 \cdot x_3 + \mathbf{0} \cdot \boldsymbol{x_4} + 1150 \cdot x_5 + 780 \cdot x_6 + 800 \cdot x_7 + 1200 \cdot x_8 + 750 \cdot x_9 + 1200 \cdot x_{10}$$

$$\boldsymbol{x_4} = \mathbf{1}$$

- Περιορισμοί:

$$x1 + x2 \geq 1$$
$$\cancel{x1 + x2 + x4 \geq 1}$$
$$\cancel{x2 + x3 + x4 \geq 1}$$
$$\cancel{x3 + x4 + x5 + x9 \geq 1}$$
$$x3 + x5 + x8 \geq 1$$
$$x3 + x5 + x6 \geq 1$$
$$x6 + x7 + x8 \geq 1$$
$$x5 + x7 + x8 + x9 + x10 \geq 1$$
$$\cancel{x4 + x9 \geq 1}$$
$$x8 + x10 \geq 1$$

$S1$: {1,2}
$S2$: {1,2,3}
$S3$: {3,4,5,6}
$S4$: {2,3,4,9}
$S5$: {4,5,6,8}
$S6$: {6,7}
$S7$: {7,8}
$S8$: {5,7,8,10}
$S9$: {4,8,9}
$S10$: {8,10}

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Βήμα 2

Δεν υπάρχει υποψήφιος που να μην εμφανίζεται σε κανέναν περιορισμό, επομένως προχωράμε άμεσα στο Βήμα 3

- Περιορισμοί:

$$x1 + x2 \geq 1$$
$$x3 + x5 + x8 \geq 1$$
$$x3 + x5 + x6 \geq 1$$
$$x6 + x7 + x8 \geq 1$$
$$x5 + x7 + x8 + x9 + x10 \geq 1$$
$$x8 + x10 \geq 1$$

$S1$: {1,2}
$S2$: {1,2,3}
$S3$: {3,4,5,6}
$S4$: {2,3,4,9}
$S5$: {4,5,6,8}
$S6$: {6,7}
$S7$: {7,8}
$S8$: {5,7,8,10}
$S9$: {4,8,9}
$S10$: {8,10}

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Βήμα 3 – Επανάληψη 1

Υπολογισμός των τιμών $v_S = \frac{c_S}{d_S}$

$$1000 \cdot x_1 + 700 \cdot x_2 + 800 \cdot x_3 + \mathbf{0} \cdot \boldsymbol{x_4} + 1150 \cdot x_5 + 780 \cdot x_6 + 800 \cdot x_7 + 1200 \cdot x_8 + 750 \cdot x_9 + 1200 \cdot x_{10}$$

$$v_1 = 1000/1 = 1000$$
$$v_2 = 700/1 = 700$$
$$v_3 = 800/2 = 400$$
$$v_5 = 1150/3 = 383.3$$
$$v_6 = 780/2 = 390$$
$$v_7 = 800/2 = 400$$
$$\boldsymbol{v_8 = 1200/4 = 300}$$
$$v_9 = 750/1 = 750$$
$$v_{10} = 1200/2 = 600$$

- Περιορισμοί:

$$x1 + x2 \geq 1$$
$$x3 + x5 + x8 \geq 1$$
$$x3 + x5 + x6 \geq 1$$
$$x6 + x7 + x8 \geq 1$$
$$x5 + x7 + x8 + x9 + x10 \geq 1$$
$$x8 + x10 \geq 1$$

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Βήμα 3 – Επανάληψη 1
  Επομένως επιλέγουμε τον 8, $\boldsymbol{x_8 = 1}$
  Διαγράφουμε όλους τους περιορισμούς που εμπλέκουν τον 8

$$x1 + x2 \geq 1$$
$$\cancel{x3 + x5 + x8 \geq 1}$$
$$x3 + x5 + x6 \geq 1$$
$$\cancel{x6 + x7 + x8 \geq 1}$$
$$\cancel{x5 + x7 + x8 + x9 + x10 \geq 1}$$
$$\cancel{x8 + x10 \geq 1}$$

Για κάθε υποψήφιο δεν εμφανίζεται πλέον στους περιορισμούς, θέτουμε $x_s = 0$,
Επομένως,

$$x_7 = 0, x_9 = 0, x_{10} = 0$$

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Βήμα 3 – Επανάληψη 2

Υπολογισμός των τιμών $v_S = \frac{c_S}{d_S}$

$$1000 \cdot x_1 + 700 \cdot x_2 + 800 \cdot x_3 + \mathbf{0} \cdot \boldsymbol{x_4} + 1150 \cdot x_5 + 780 \cdot x_6 + 800 \cdot x_7 + \mathbf{1200} \cdot \boldsymbol{x_8} + 750 \cdot x_9 + 1200 \cdot x_{10}$$

$$v_1 = 1000/1 = 1000$$
$$\boldsymbol{v_2 = 700/1 = 700}$$
$$v_3 = 800/1 = 800$$
$$v_5 = 1150/1 = 1150$$
$$v_6 = 780/1 = 780$$

- Περιορισμοί:

$$x1 + x2 \geq 1$$
$$x3 + x5 + x6 \geq 1$$

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Βήμα 3 – Επανάληψη 2
  Επομένως επιλέγουμε τον 2, $\boldsymbol{x_2 = 1}$
  Διαγράφουμε όλους τους περιορισμούς που εμπλέκουν τον 2

$$\cancel{x1 + x2 \geq 1}$$
$$x3 + x5 + x6 \geq 1$$

Για κάθε υποψήφιο δεν εμφανίζεται πλέον στους περιορισμούς, θέτουμε $x_s = 0$,
Επομένως,

$$x_1 = 0$$

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Βήμα 3 – Επανάληψη 3

Υπολογισμός των τιμών $v_S = \frac{c_S}{d_S}$

$$v_3 = 800$$
$$v_5 = 1150/1 = 1150$$
$$\boldsymbol{v_6 = 780/1 = 780}$$

$$1000 \cdot x_1 + \boldsymbol{700 \cdot x_2} + 800 \cdot x_3 + \boldsymbol{0 \cdot x_4} + 1150 \cdot x_5 + 780 \cdot x_6 + 800 \cdot x_7 + \boldsymbol{1200 \cdot x_8} + 750 \cdot x_9$$

- Περιορισμοί:

$$x3 + x5 + x6 \geq 1$$

ΤΟ ΠΡΟΒΛΗΜΑ Weighted SET COVERING

- Βήμα 3 – Επανάληψη 2
  Επομένως επιλέγουμε τον 6, $\boldsymbol{x_6 = 1}$
  Διαγράφουμε όλους τους περιορισμούς που εμπλέκουν τον 6

  $$\cancel{x3 + x5 + x6 \geq 1}$$

  Τερματισμός Αλγορίθμου (Δεν υπάρχουν άλλοι περιορισμοί, δηλαδή δεν υπάρχουν δεξιότητες που δεν έχουν καλυφθεί)

- Τελική λύση:

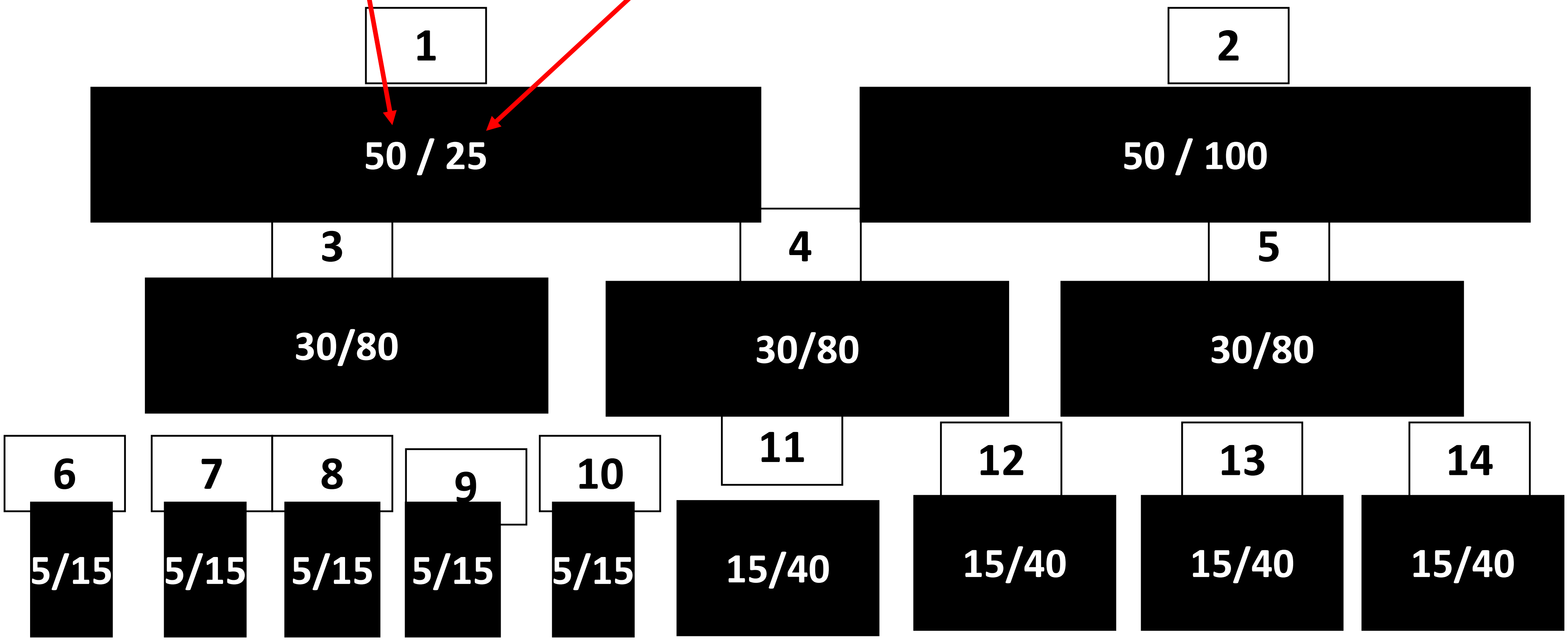
  $$s = \{S_2, S_4, S_6, S_8\}$$

  Με αντικειμενική συνάρτηση:

  $$z(s) = 700 + 0 + 780 + 1200 = 2680$$

Πρόβλημα Knapsack

Δεδομένου ενός συνόλου 14 containers, καθένα από τα οποία χαρακτηρίζεται από ένα συγκεκριμένο βάρος και μια συγκεκριμένη αξία, καθόρισε τον αριθμό των αντικειμένων που θα επιλεγούν να μεταφερθούν με ένα φορτηγό πλοίο που αντέχει συνολικό βάρος μέχρι και 120 μονάδες βάρους. Στόχος του προβλήματος είναι να μεγιστοποιηθεί η συνολική αξία των containers που θα μεταφερθούν με το φορτηγό πλοίο.



ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

- Το συγκεκριμένο επιχειρησιακό σενάριο αποτελεί εφαρμογή του προβλήματος Knapsack
- <u>Μορφή Λύσης</u>: Ένα σύνολο από επιλεγμένα containers.
- <u>Στοιχείο Λύσης</u>: Ένα container που προστίθεται στην ημιτελή λύση.
- <u>Κριτήριο Επιλογής</u>: Υπολόγισε το κλάσμα (αξία/μέγεθος) για κάθε container και κατάταξε τα containers σε φθίνουσα σειρά του κλάσματος (αξία/μέγεθος)

  Ξεκινώντας από το πρώτο container, έλεγξε αν είναι εφικτή η πρόσθεση του container στη λύση. Εφόσον είναι κάνε την συγκεκριμένη προσθήκη.

  Σε κάθε επανάληψη, επανάλαβε τα ίδια βήματα για το επόμενο container

ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

–Επανάληψη 1. Επειδή περισσότερα από ένα containers ικανοποιούν με τον καλύτερο δυνατό τρόπο το «Κριτήριο Επιλογής» (το μικρότερο κλάσμα για την επανάληψη 1 έχουν τα containers 6,7,8,9 και 10), επιλέγουμε στοχαστικά ένα από αυτά. Έστω ότι επιλέγουμε στοχαστικά το container 6, τότε η ημιτελής λύση S διαμορφώνεται ως S:{6}.

–Επανάληψη 2. Ομοίως, S:{6,10}.

–Επανάληψη 3. Ομοίως, S:{6,10,7}.

–Επανάληψη 4. Ομοίως, S:{6,10,7,8}.

–Επανάληψη 5. Ομοίως, S:{6,10,7,8,9}.

ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

– Επανάληψη 6. Ομοίως, S:{6,10,7,8,9,11}.

– Επανάληψη 7. Ομοίως, S:{6,10,7,8,9,11,12}.

– Επανάληψη 8. Ομοίως, S:{6,10,7,8,9,11,12,13}.

– Επανάληψη 9. Ομοίως, S:{6,10,7,8,9,11,12,13,14}.

– Επανάληψη 10. Ομοίως, S:{6,10,7,8,9,11,12,13,14,3}.

Η συνολική αξία των επιλεγμένων αντικειμένων είναι ίση με

P(s) = 15+15+15+15+15+40+40+40+40+80 = 315

ΣΤΡΑΤΗΓΙΚΟ ΠΡΟΒΛΗΜΑ

- Έχουμε στην κατοχή μας ένα οδικό δίκτυο, το οποίο αποτελείται από διάφορες τοποθεσίες
- Χρειάζεται να χαράξουμε το δρομολόγιο ενός λεωφορείου με τέτοιο τρόπο, ώστε να φτάνει από μία τοποθεσία A σε μία τοποθεσία B στο μικρότερο δυνατό χρονικό διάστημα
- Δεν είναι ανάγκη να επισκεφθούμε όλους του κόμβους του δικτύου
- Κάθε ζεύγος κόμβων ενώνεται με ένα τόξο
- Κάθε τόξο χαρακτηρίζεται από το χρόνο μετάβασης μεταξύ των δύο εμπλεκομένων τοποθεσιών

ΠΡΟΒΛΗΜΑ

- Ποια είναι η μορφή της λύσης;
- Πως πρέπει να εργαστούμε για να λύσουμε το πρόβλημα;

ΠΡΟΒΛΗΜΑΤΑ ΣΥΝΤΟΜΟΤΕΡΟΥ ΜΟΝΟΠΑΤΙΟΥ

- Το συγκεκριμένο πρόβλημα είναι γνωστό ως το πρόβλημα του Συντομότερου Μονοπατιού (Shortest Path)
- Είναι ένα πρόβλημα που εντάσσεται σε μία ευρύτερη κατηγορία προβλημάτων τα οποία *εξερευνούν γράφους* (graph exploring algorithms)
- Έχει κεντρική θέση στην επίλυση προβλημάτων
  - GPS Navigation
  - Κατασκευής πινάκων αποστάσεων
  - Συμμετέχει στις column generation μεθοδολογίες της Επιχειρησιακής Έρευνας
- Χαρακτηριστικότερο παράδειγμα
  - Google Maps

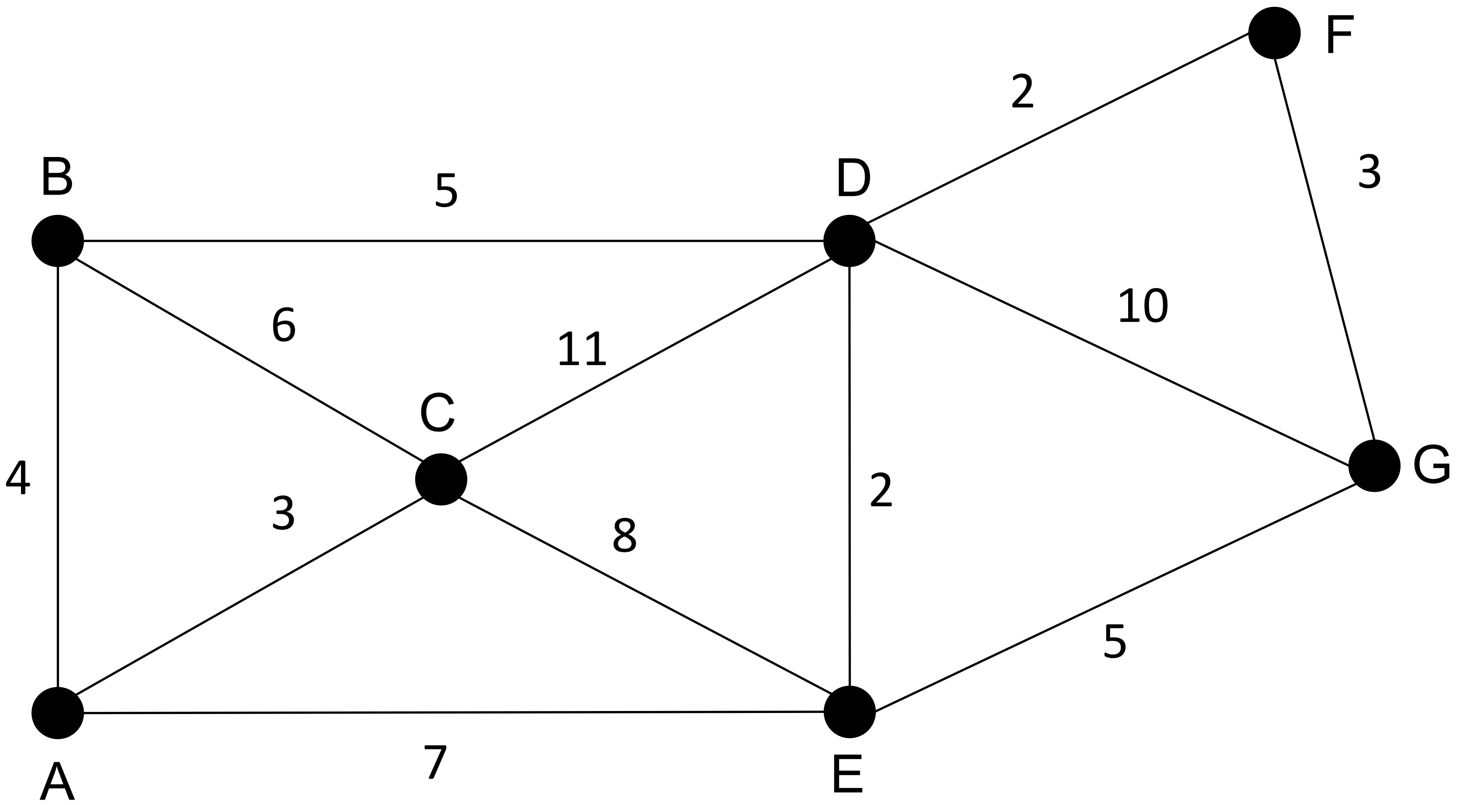
ΠΡΟΒΛΗΜΑΤΑ ΣΥΝΤΟΜΟΤΕΡΟΥ ΜΟΝΟΠΑΤΙΟΥ

- Το συγκεκριμένο πρόβλημα είναι γνωστό ως το πρόβλημα του Συντομότερου Μονοπατιού (Shortest Path)
- Είναι ένα πρόβλημα που εντάσσεται σε μία ευρύτερη κατηγορία προβλημάτων τα οποία *εξερευνούν γράφους* (graph exploring algorithms)
- Έχει κεντρική θέση στην επίλυση προβλημάτων
  - GPS Navigation
  - Κατασκευής πινάκων αποστάσεων
  - Συμμετέχει στις column generation μεθοδολογίες της Επιχειρησιακής Έρευνας
- Χαρακτηριστικότερο παράδειγμα
  - Google Maps

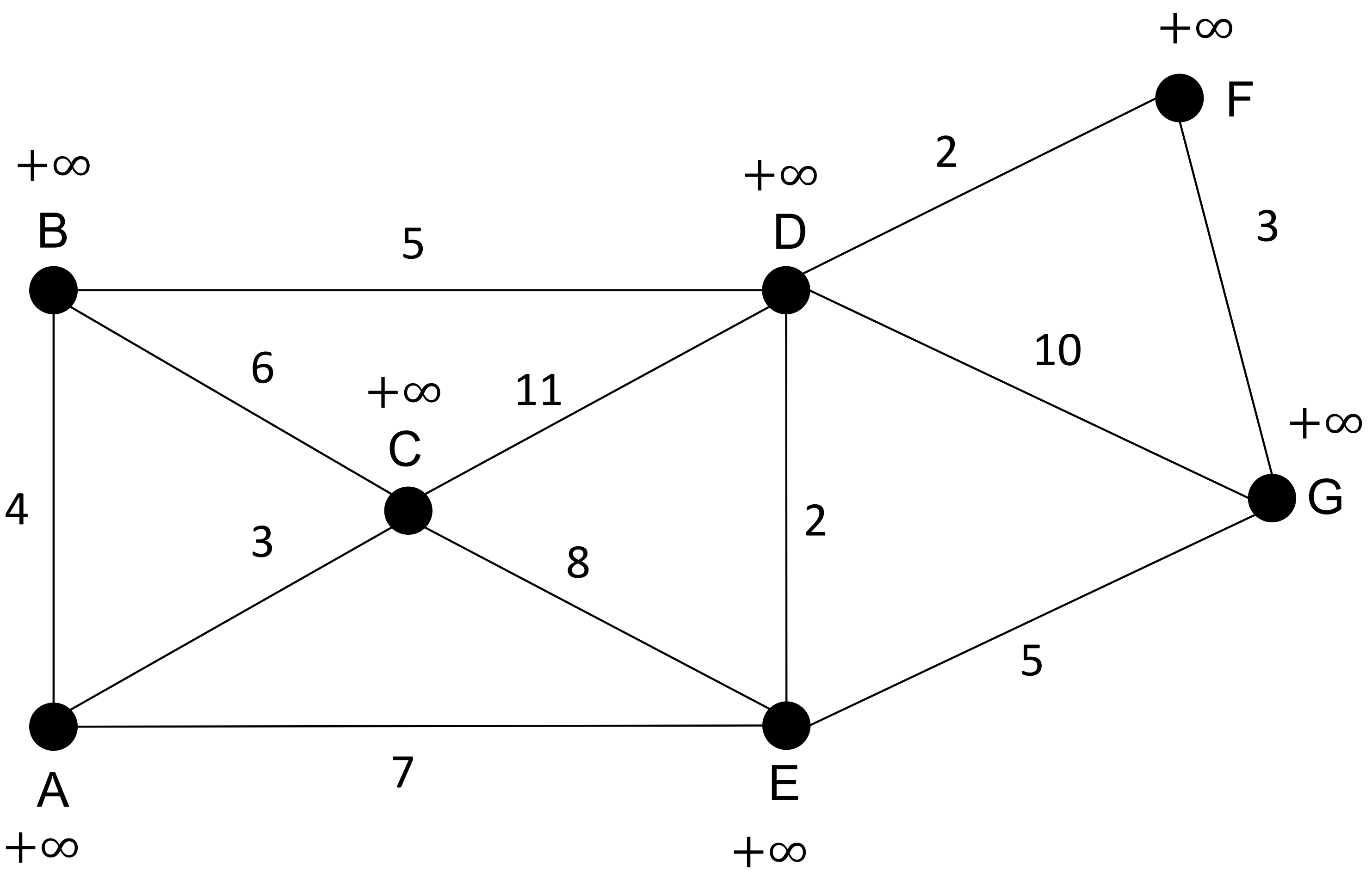
ΠΡΟΒΛΗΜΑΤΑ ΣΥΝΤΟΜΟΤΕΡΟΥ ΜΟΝΟΠΑΤΙΟΥ

- Για τα Shortest Path προβλήματα υπάρχει ένς πλεονεκτικός Αλγόριθμος ο οποίος εγγυάται την παραγωγήτης Βέλτιστης Λύσης
- Ο Αλγόριθμος αυτός ονομάζεται Αλγόριθμος του Dijkstra
- Βήματα του Αλγορίθμου
  - Αρχικοποίησε του κόμβους $\forall i \in N$ με ένα κόστος $dist_i = +\infty$
  - Θέσε το κόστος του Κόμβου Αφετηρίας s ίσο με $dist_s = 0$
  - $V \leftarrow \{s\}$
  - Επαναληπτικά και όσο ο Κόμβος Προορισμός $t \notin V$
    - Επίλεξε τον Κόμβο n που δεν ανήκει σε αυτούς που δεν έχουμε επισκεφτεί με το ελάχιστο κόστος
      - Θέσε το κόστος του Κόμβου n ίσο με το ελάχιστο αυτό κόστος
      - $V \leftarrow V \cup \{n\}$
    - Ανανέωσε το κόστος για όλους τους γειτονικούς Κόμβους (k) του n τους οποίους οποίους δεν έχουμε ακόμα επισκεφτεί ($k \notin V$)
      - Αν το κόστος μετάβασης στον k μέσω του n είναι μικρότερο από το τρέχον κόστος του k, θέσε την τιμή κόστους του γείτονα k σε αυτήν τη μικρότερη τιμή

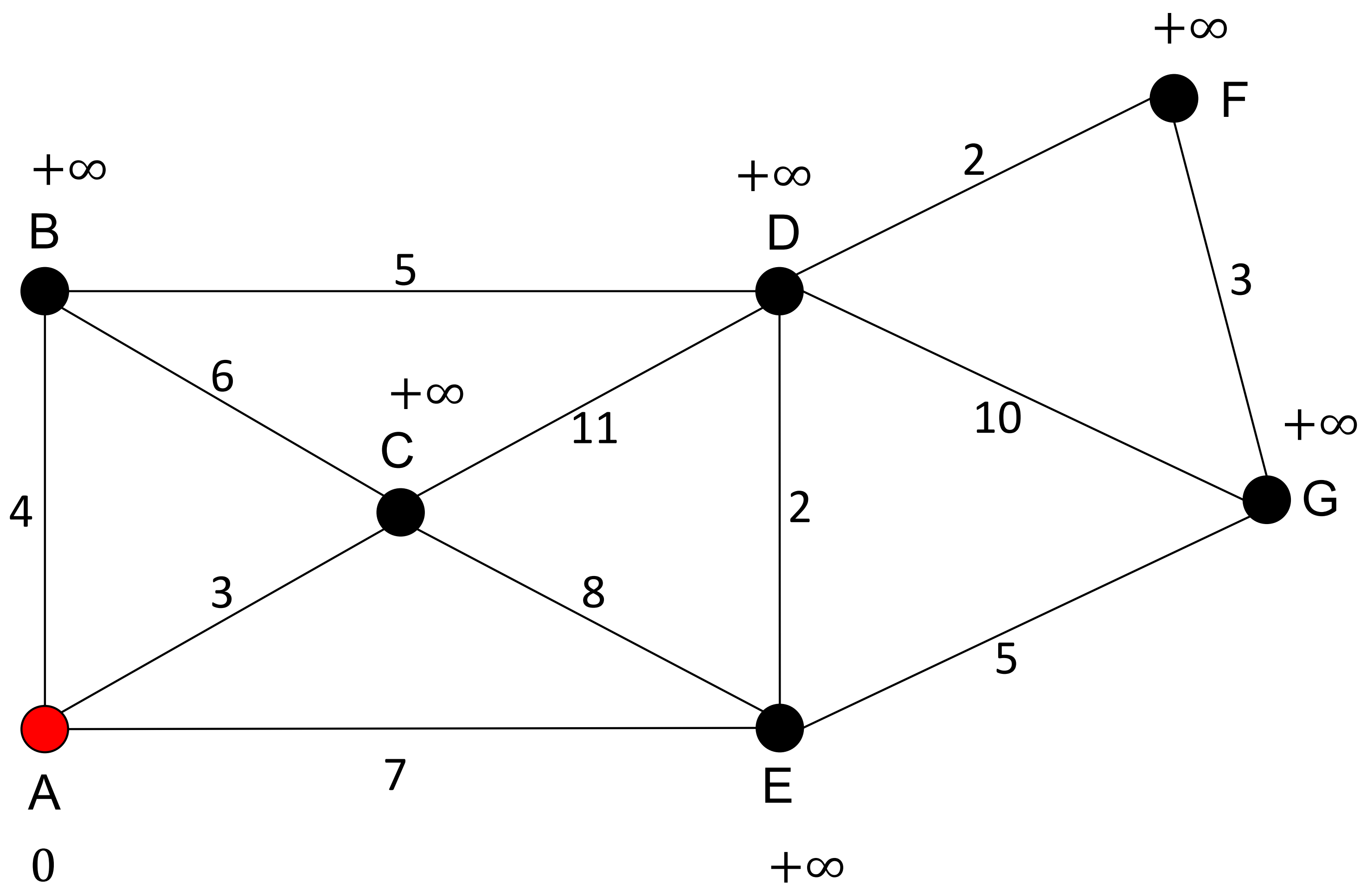
ΔΕΔΟΜΕΝΑ



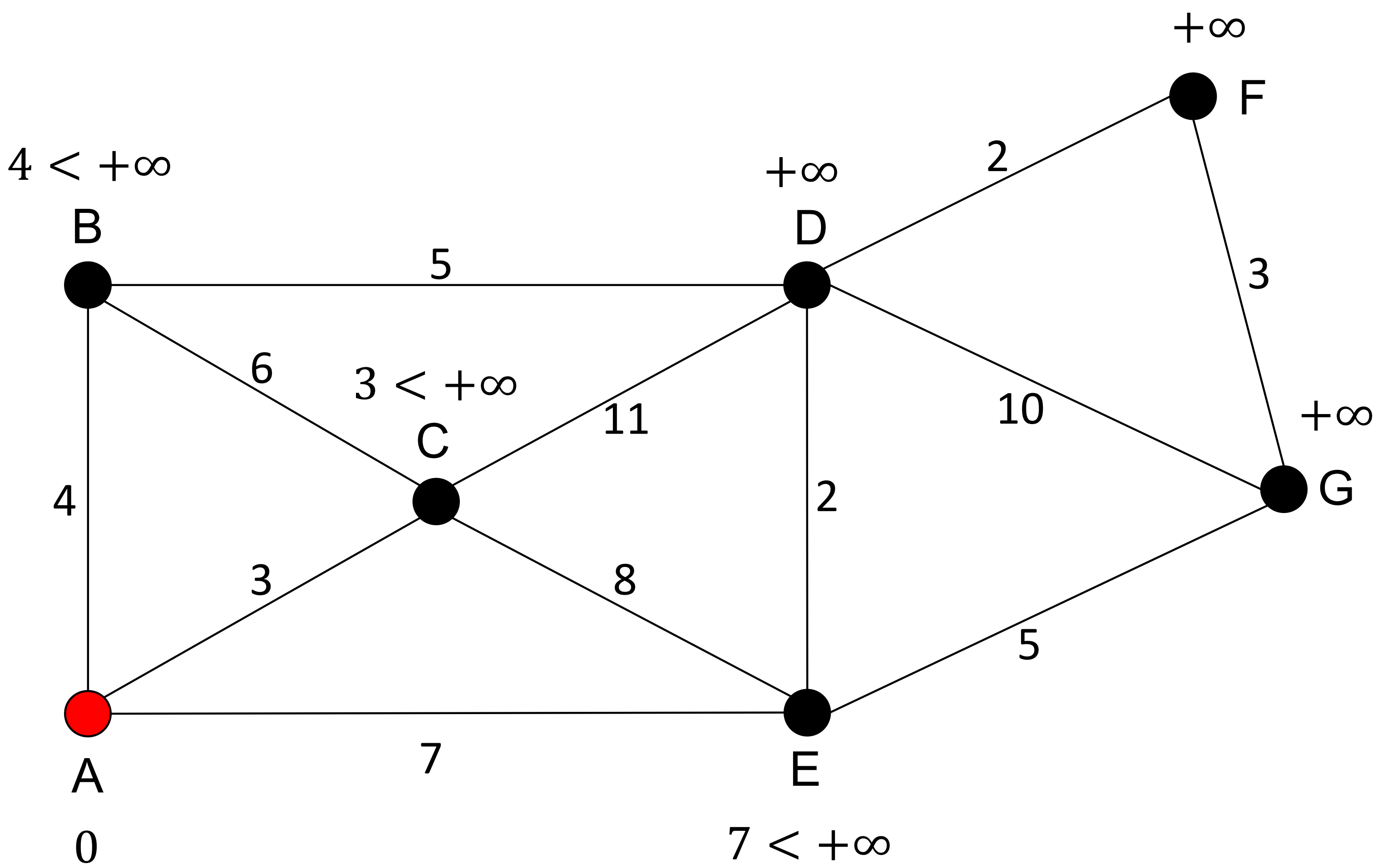
ΑΡΧΙΚΟΠΟΙΗΣΗ Dijkstra



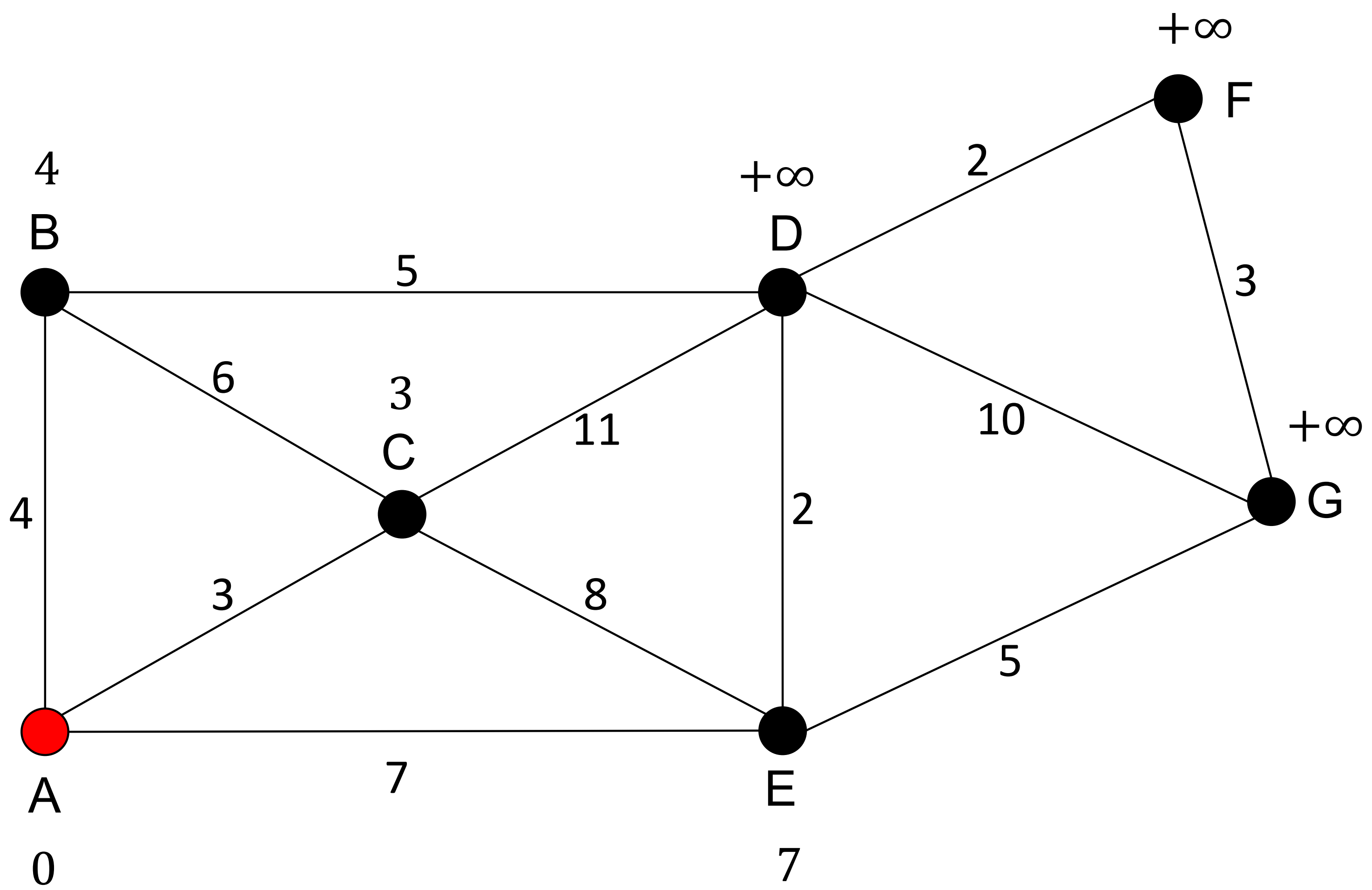
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



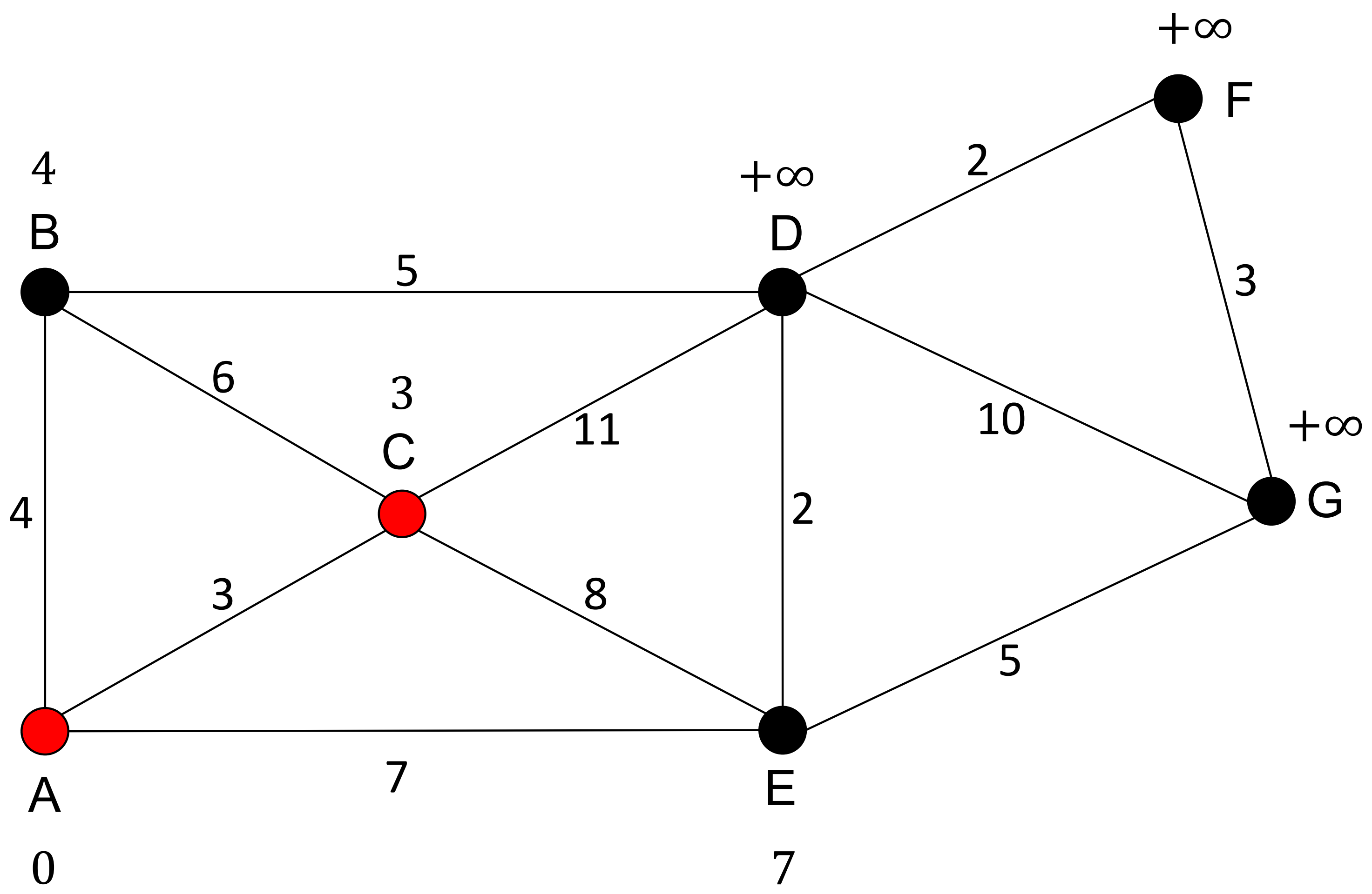
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



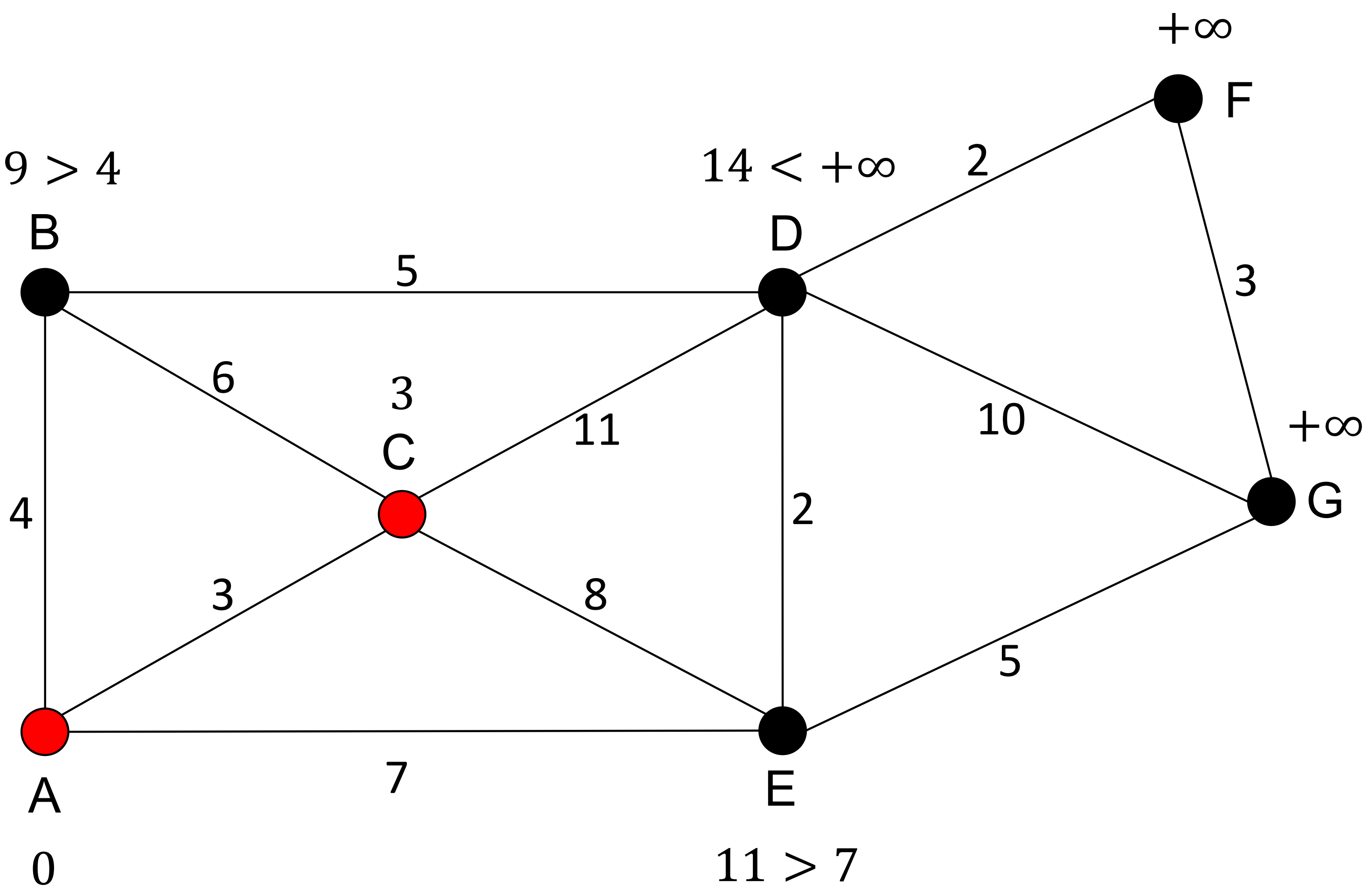
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



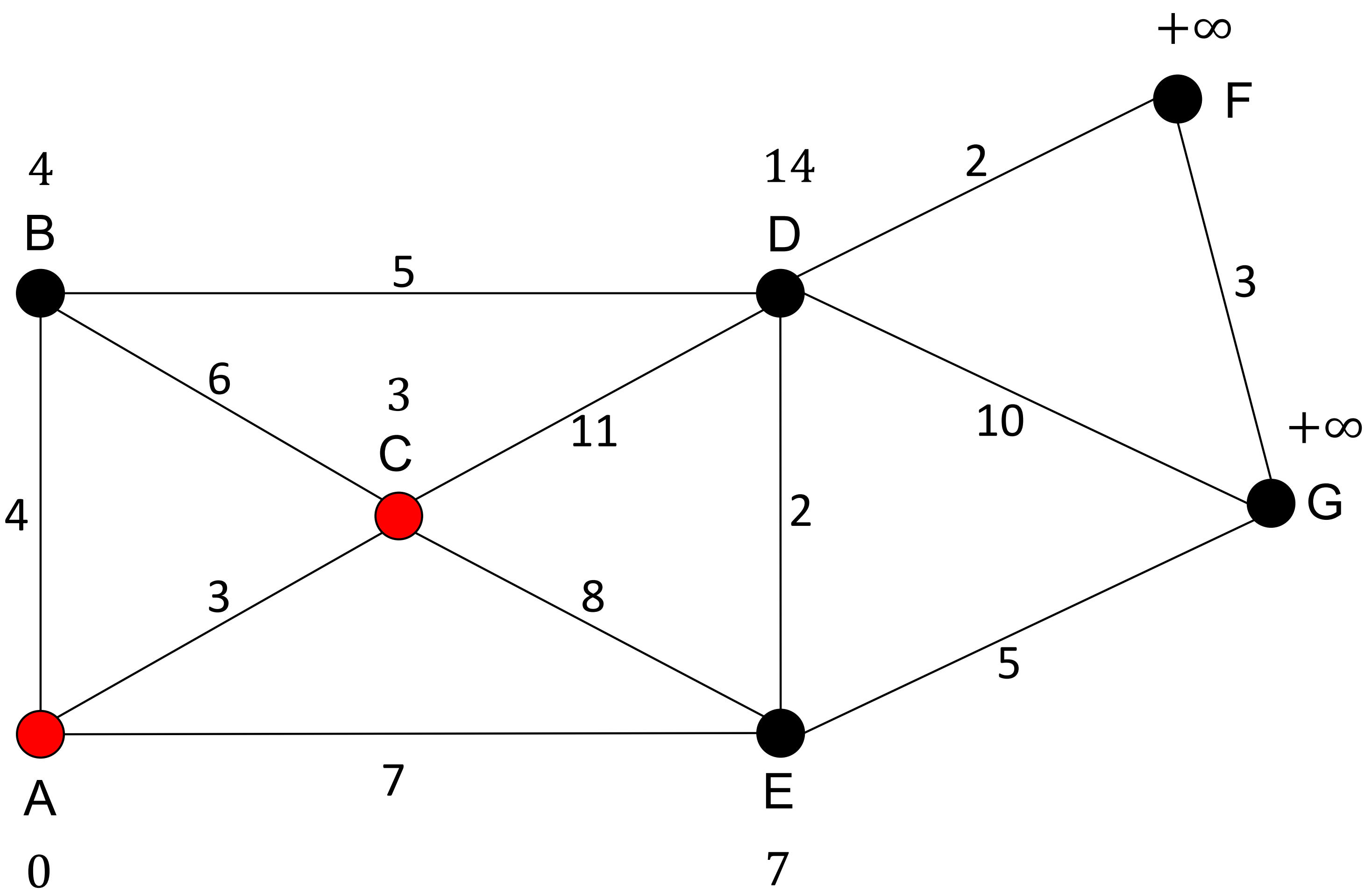
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



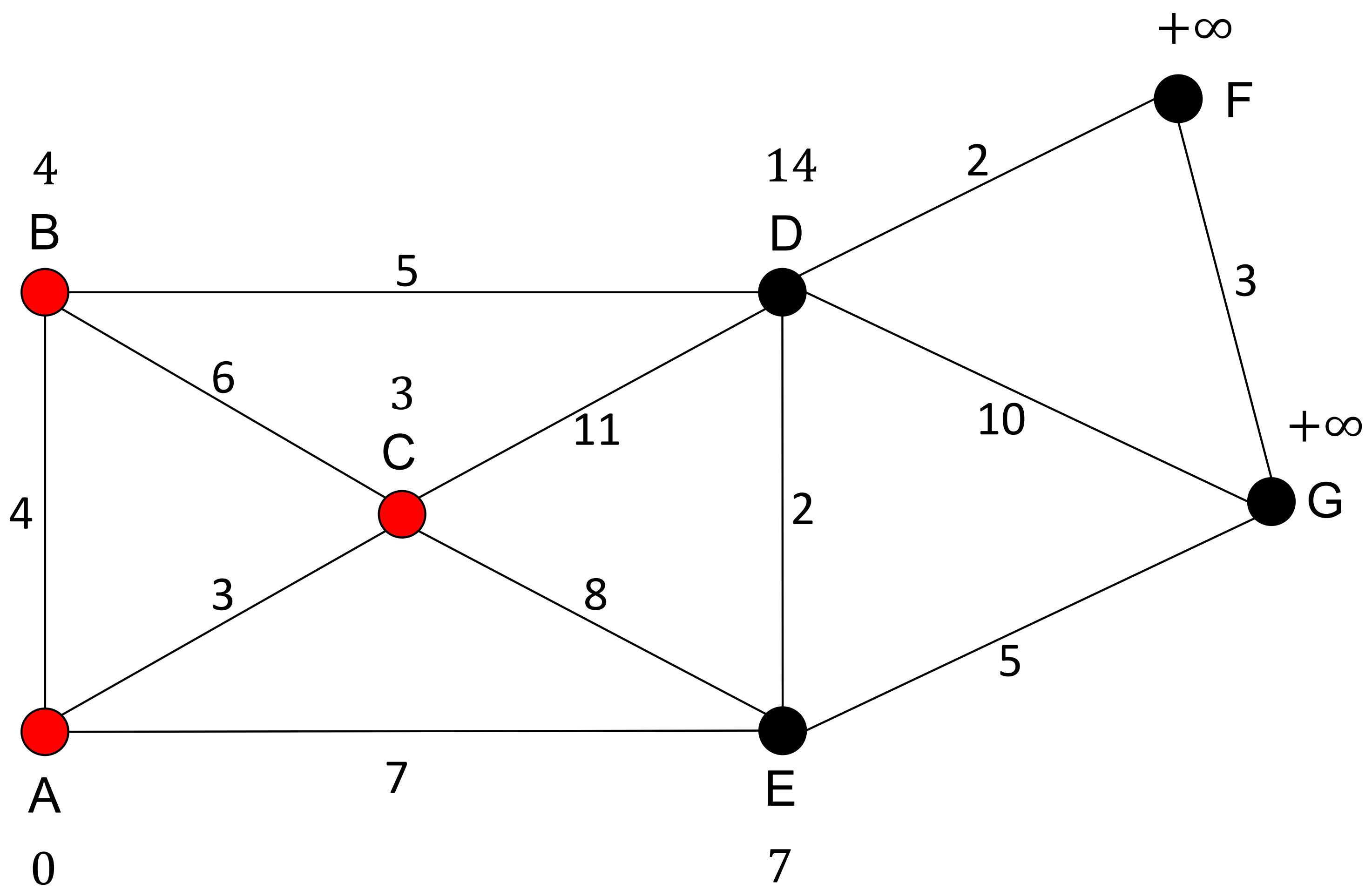
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



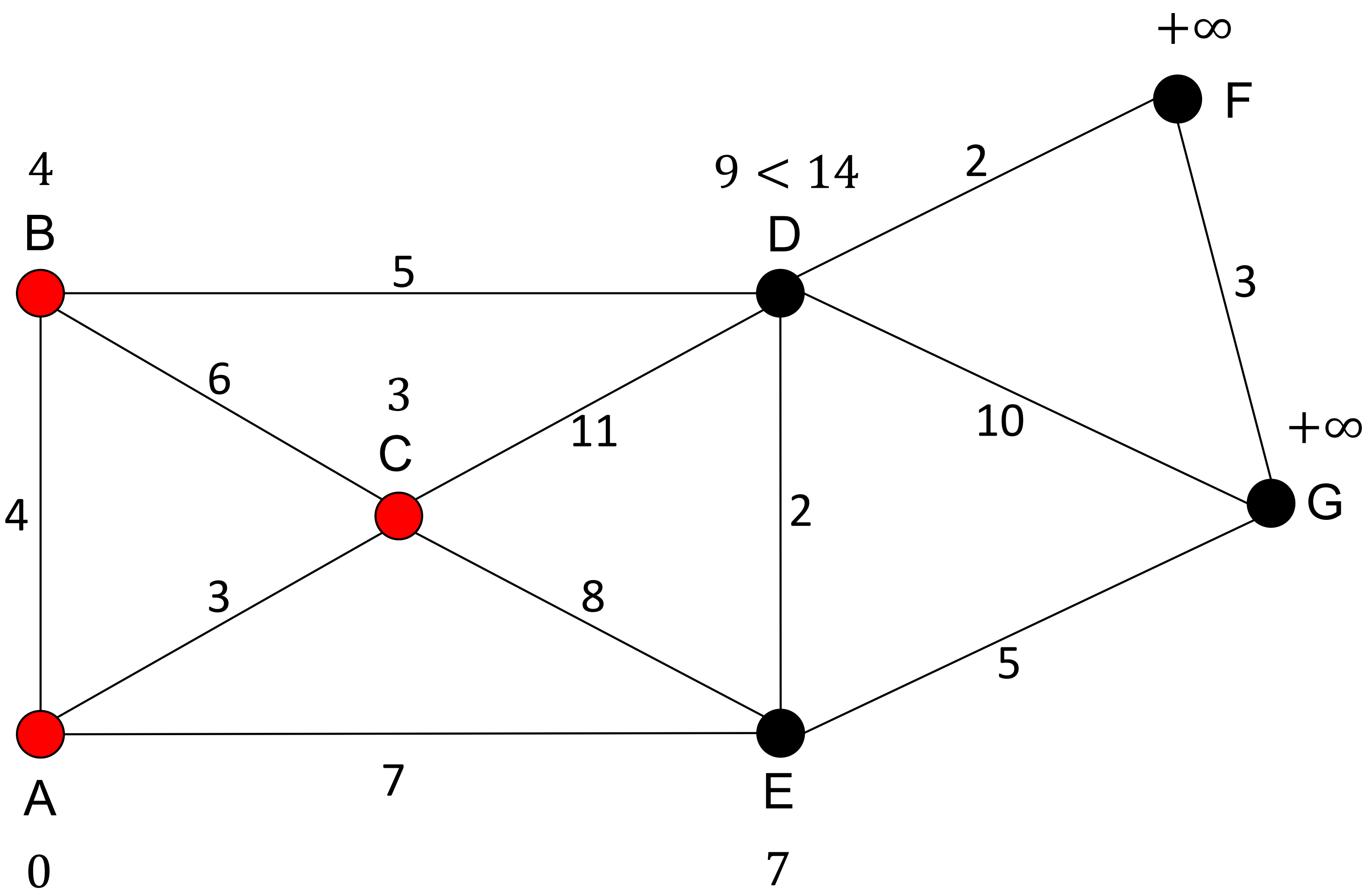
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



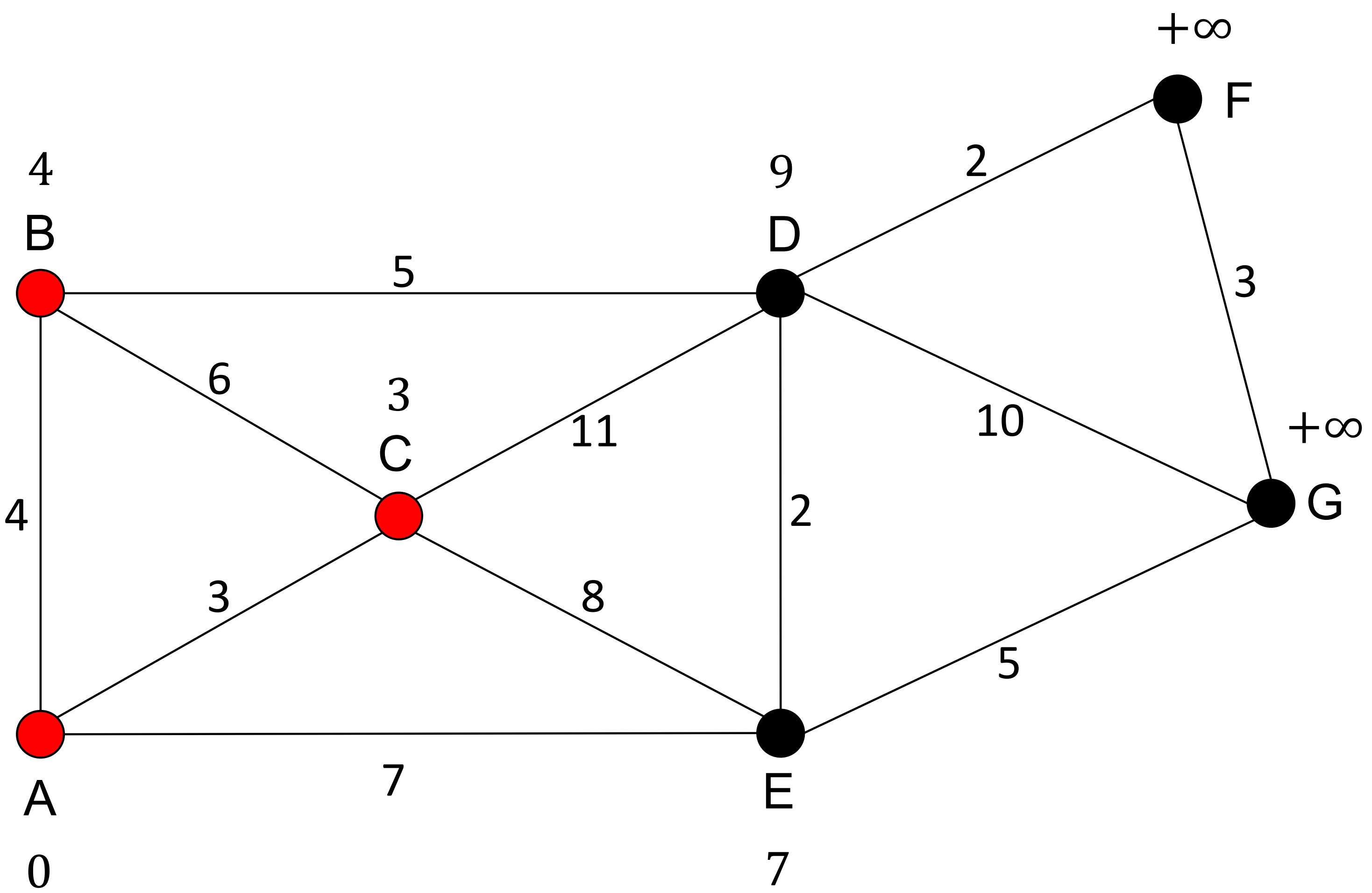
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



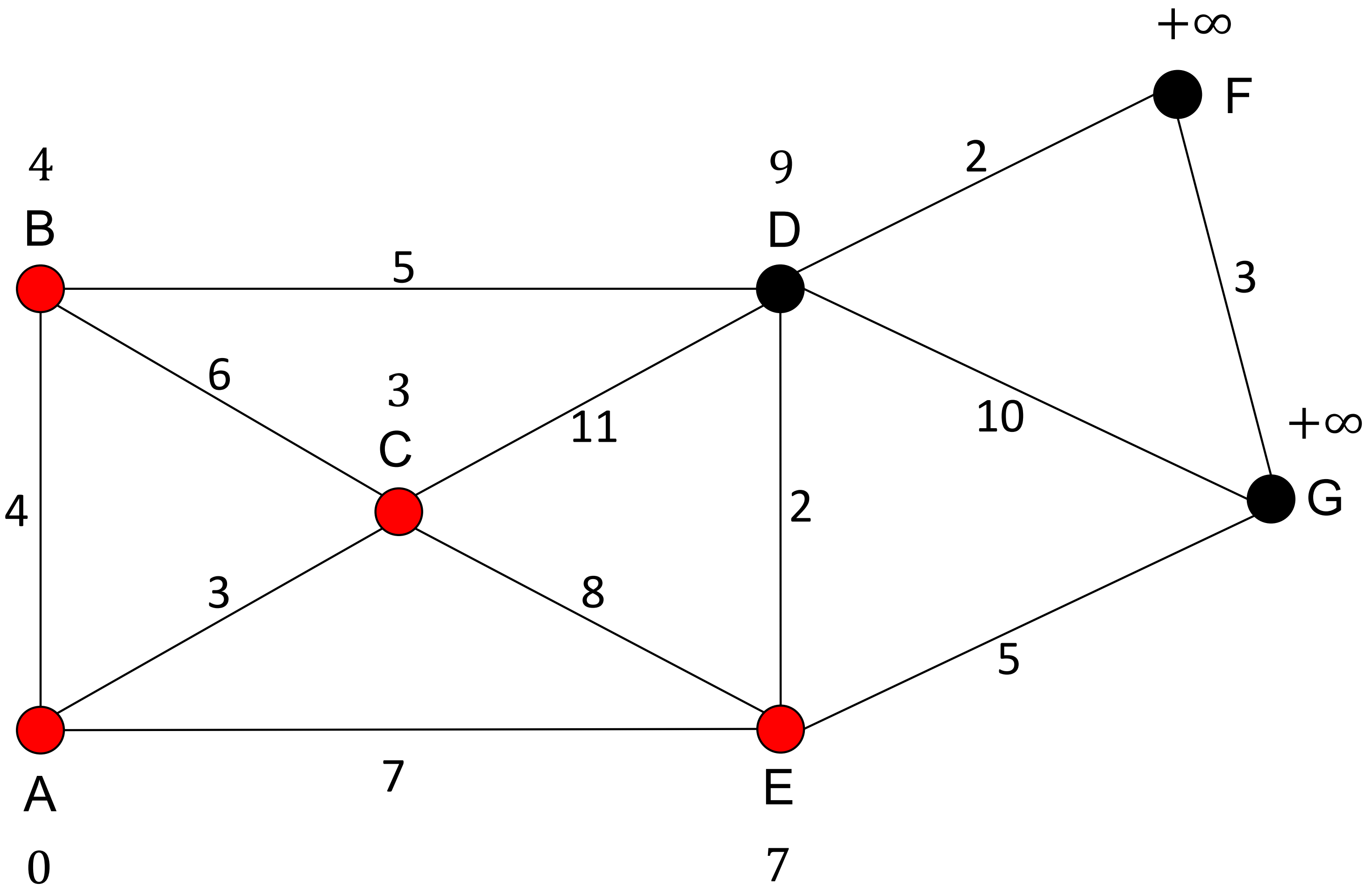
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



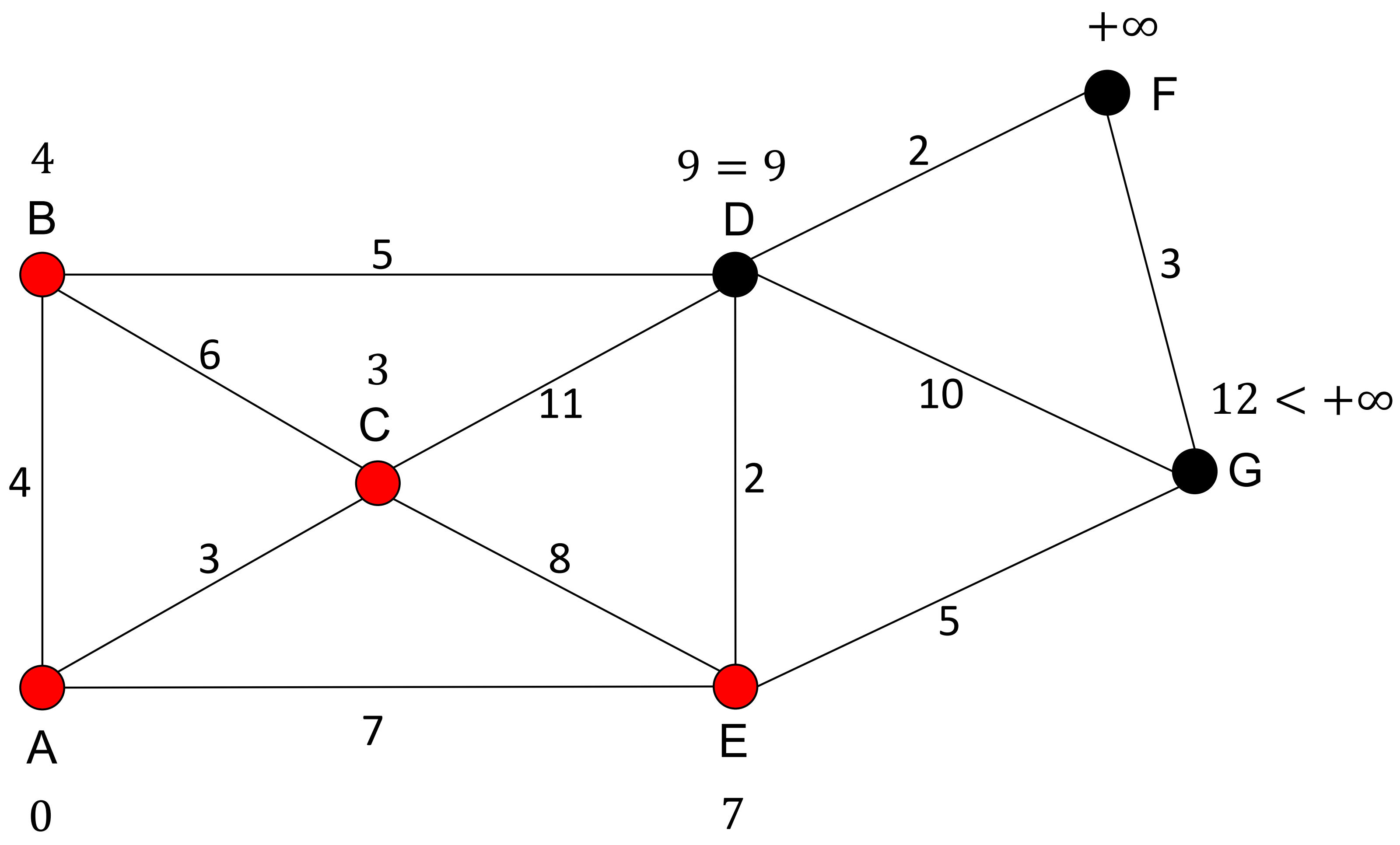
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



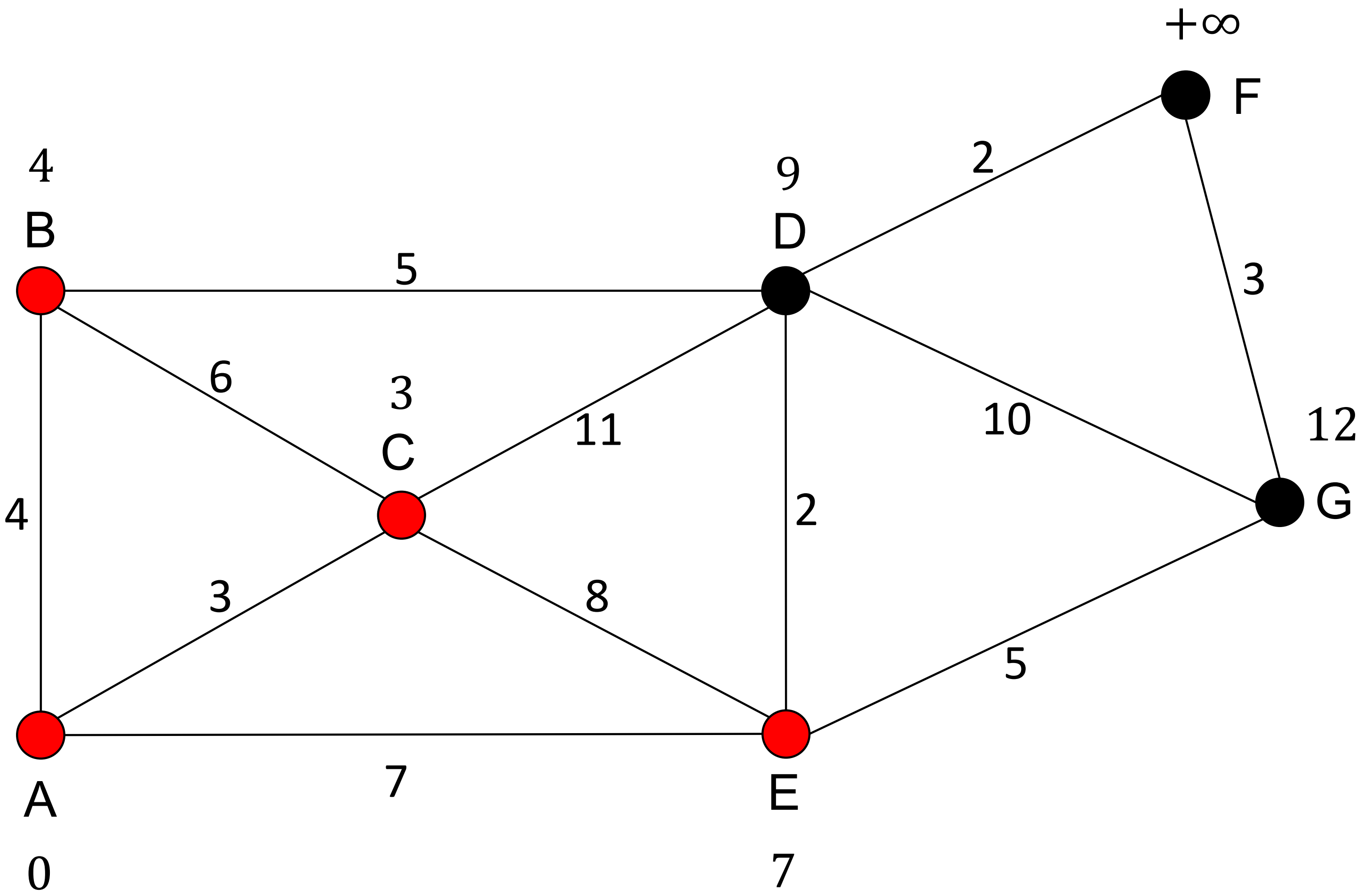
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



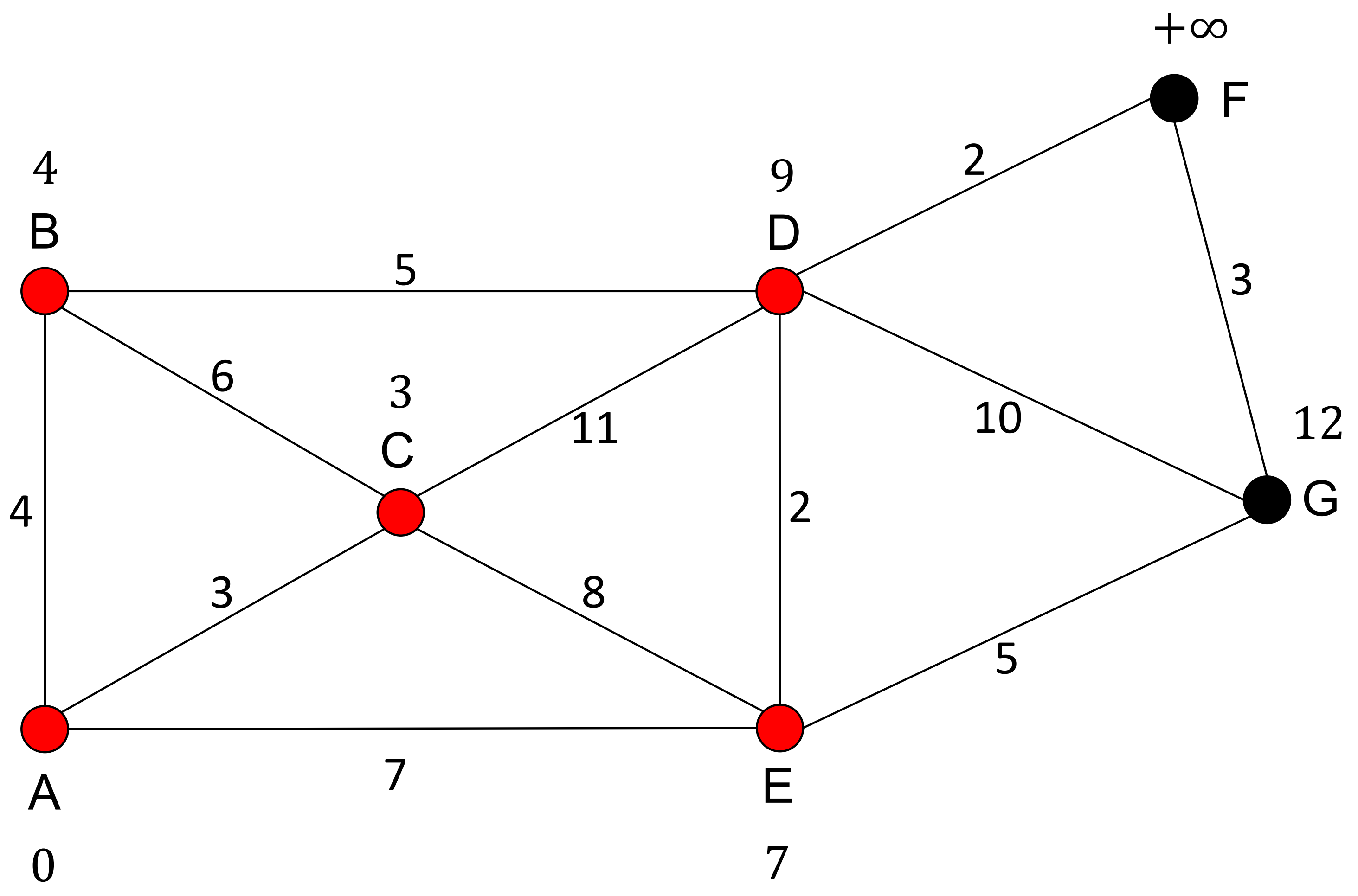
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



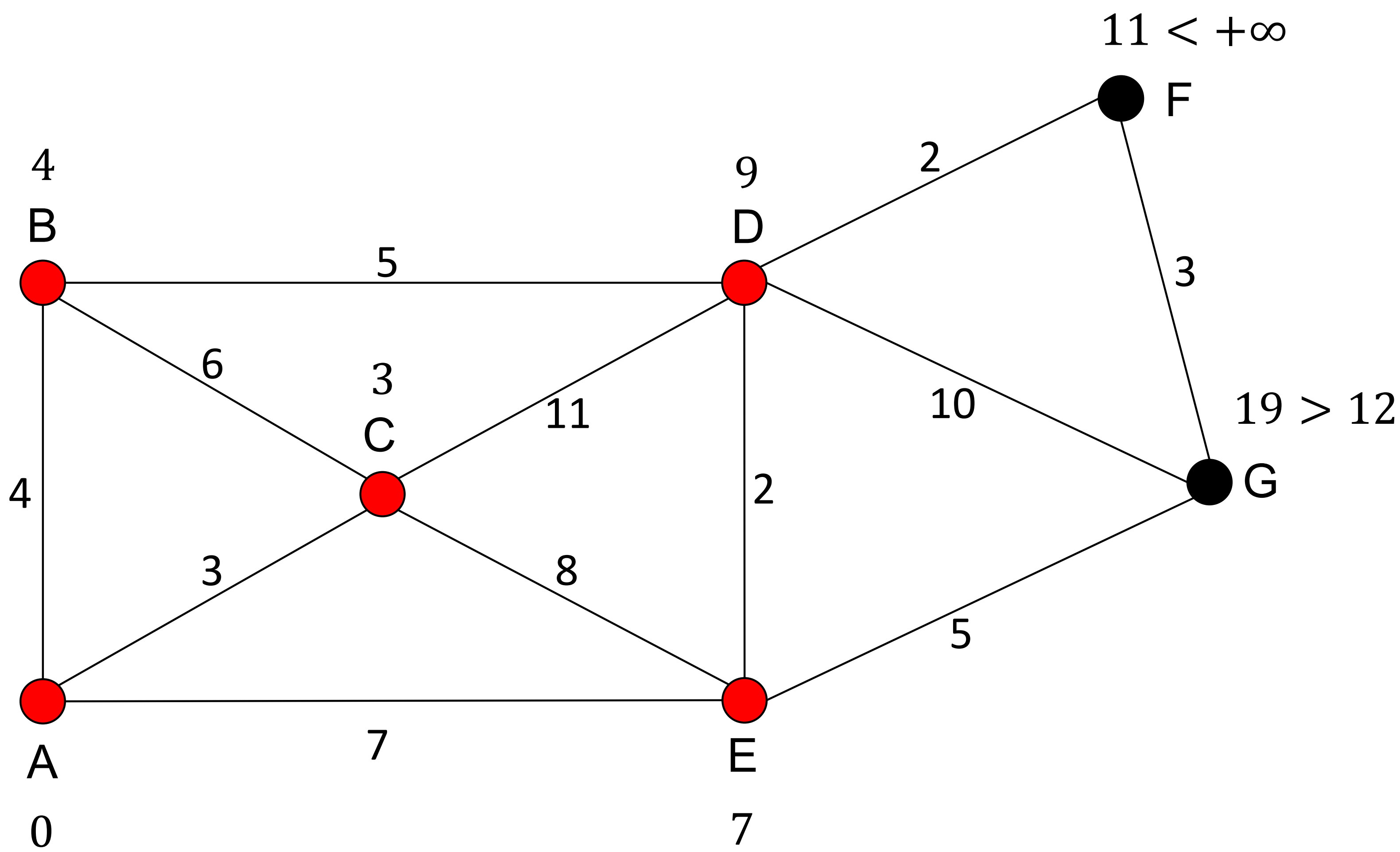
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



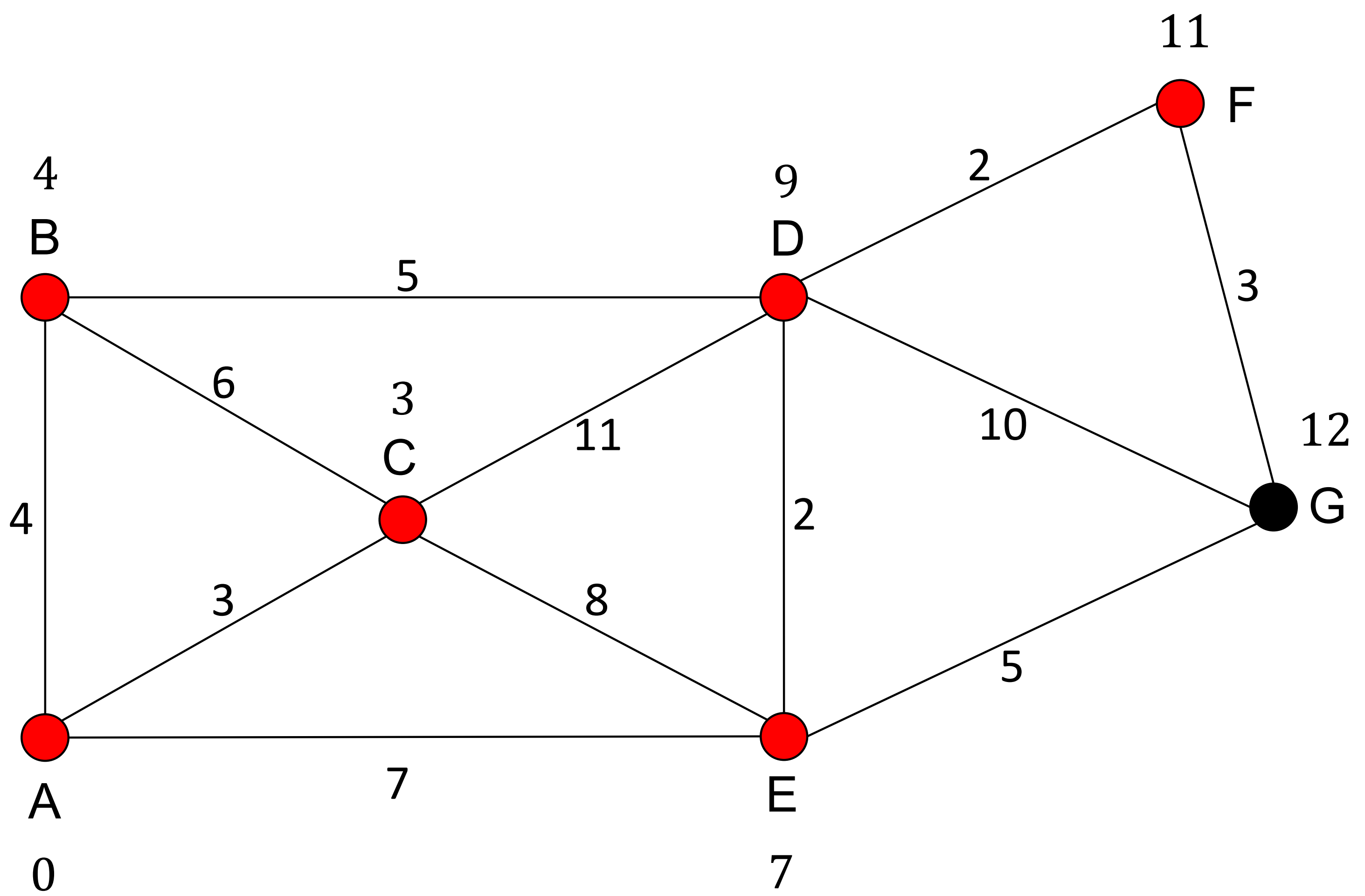
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



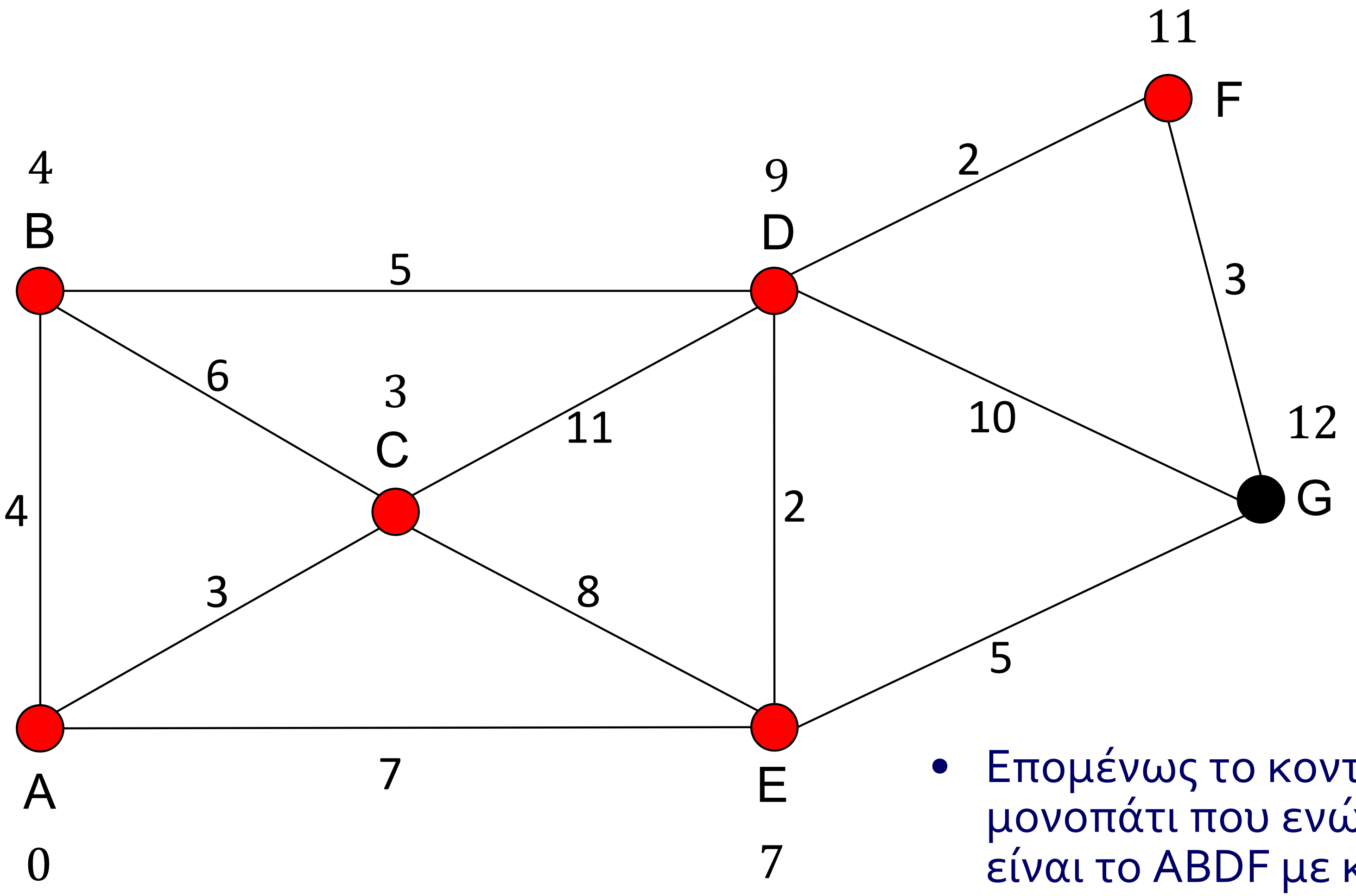
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



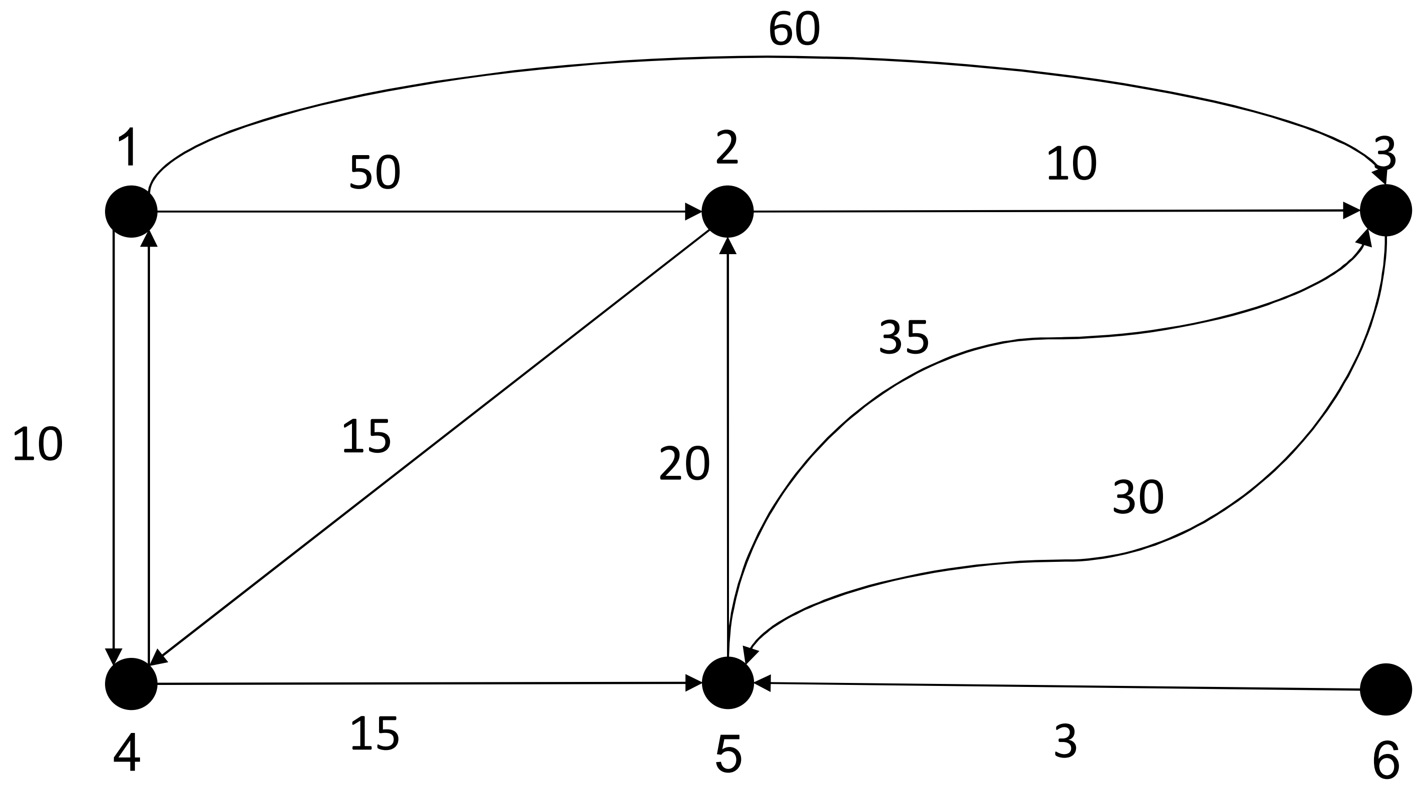
Ο ΠΛΕΟΝΕΚΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ Dijkstra



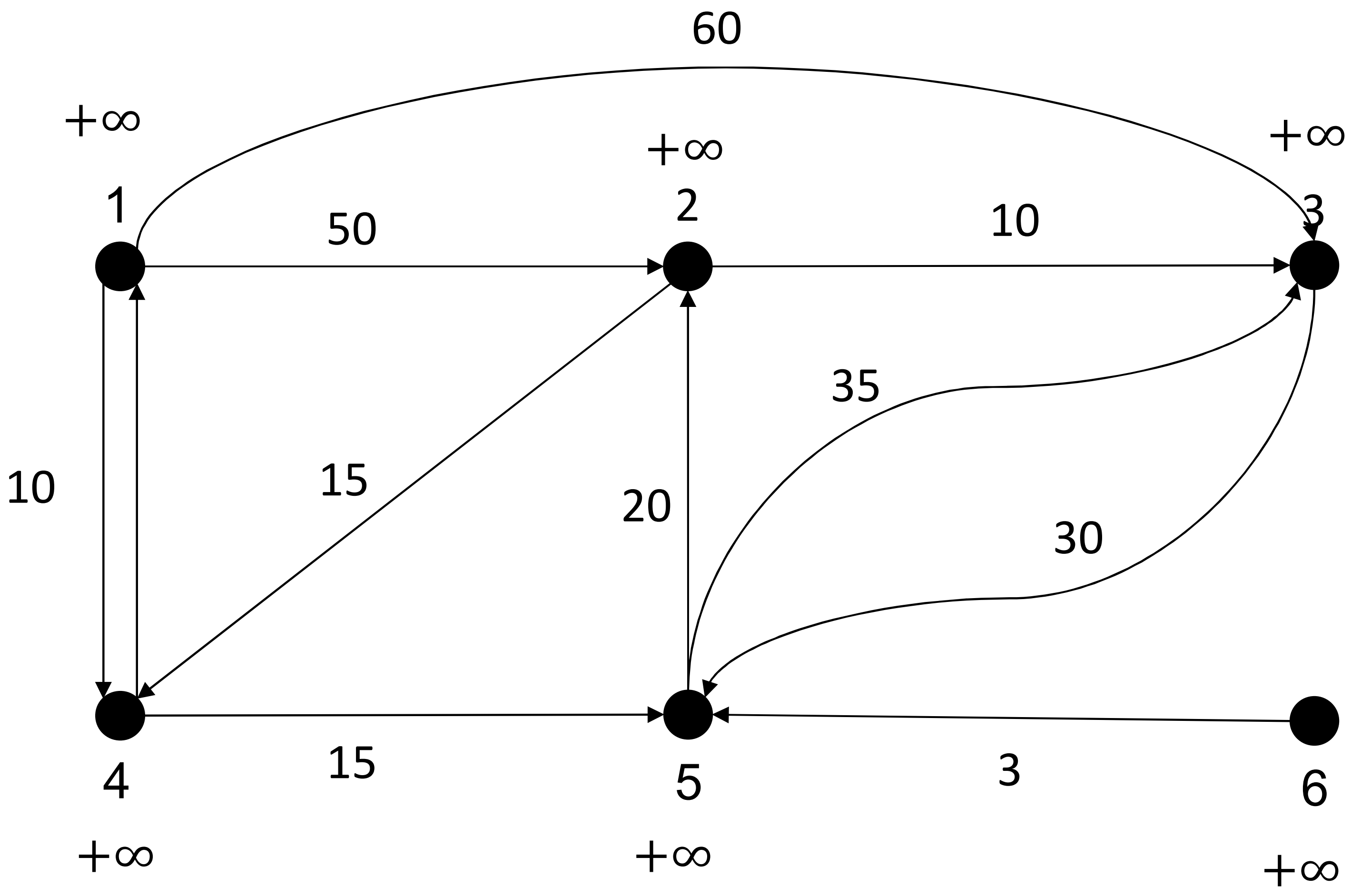
- Επομένως το κοντινότερο μονοπάτι που ενώνει τα A και F είναι το ABDF με κόστος 11

Παράδειγμα Shortest Path

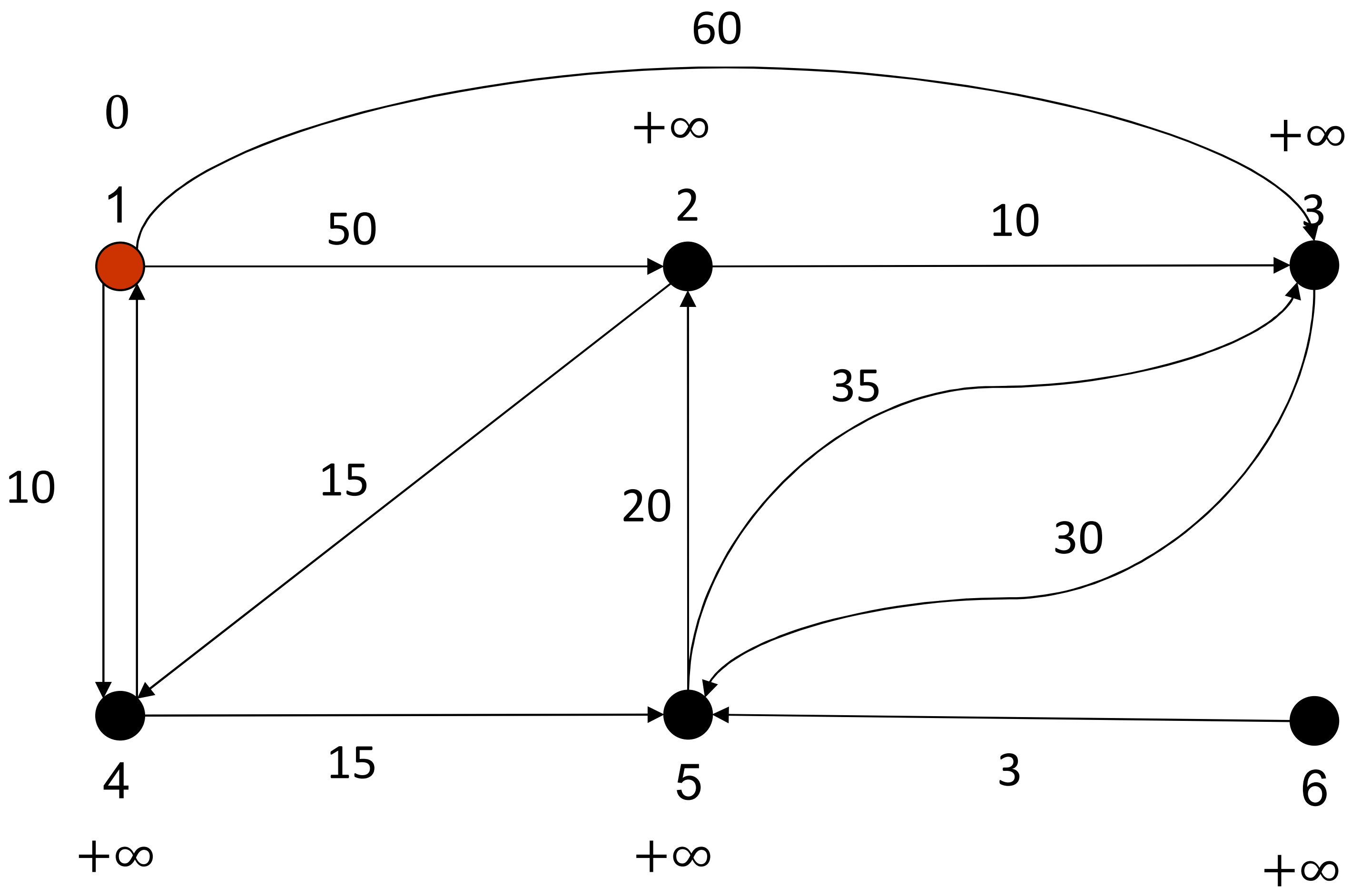
Βρείτε το συντομότερο μονοπάτι το οποίο ξεκινά από τον Κόμβο 1 και καταλήγει στον Κόμβο 3



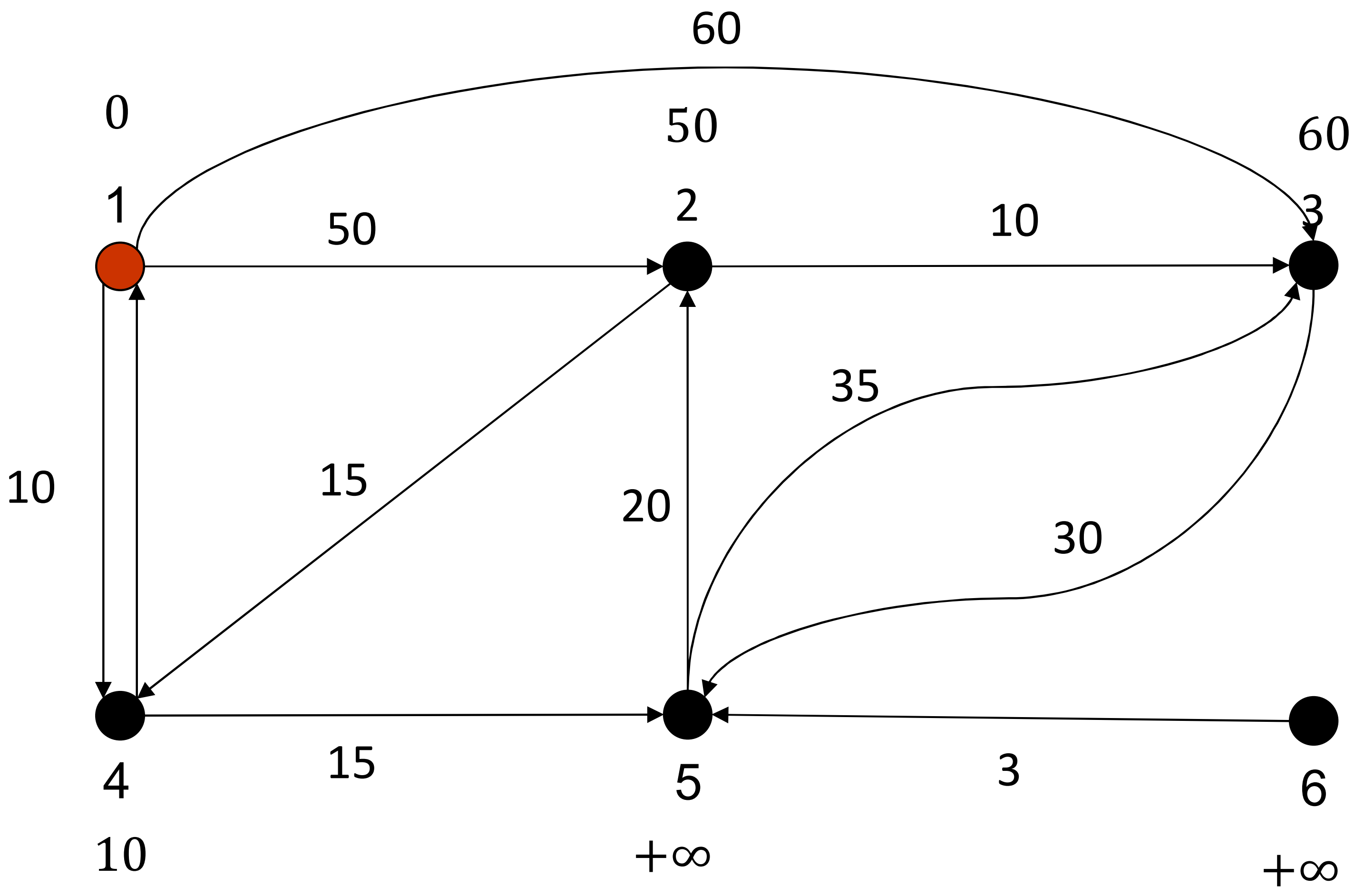
Αρχικοποίηση



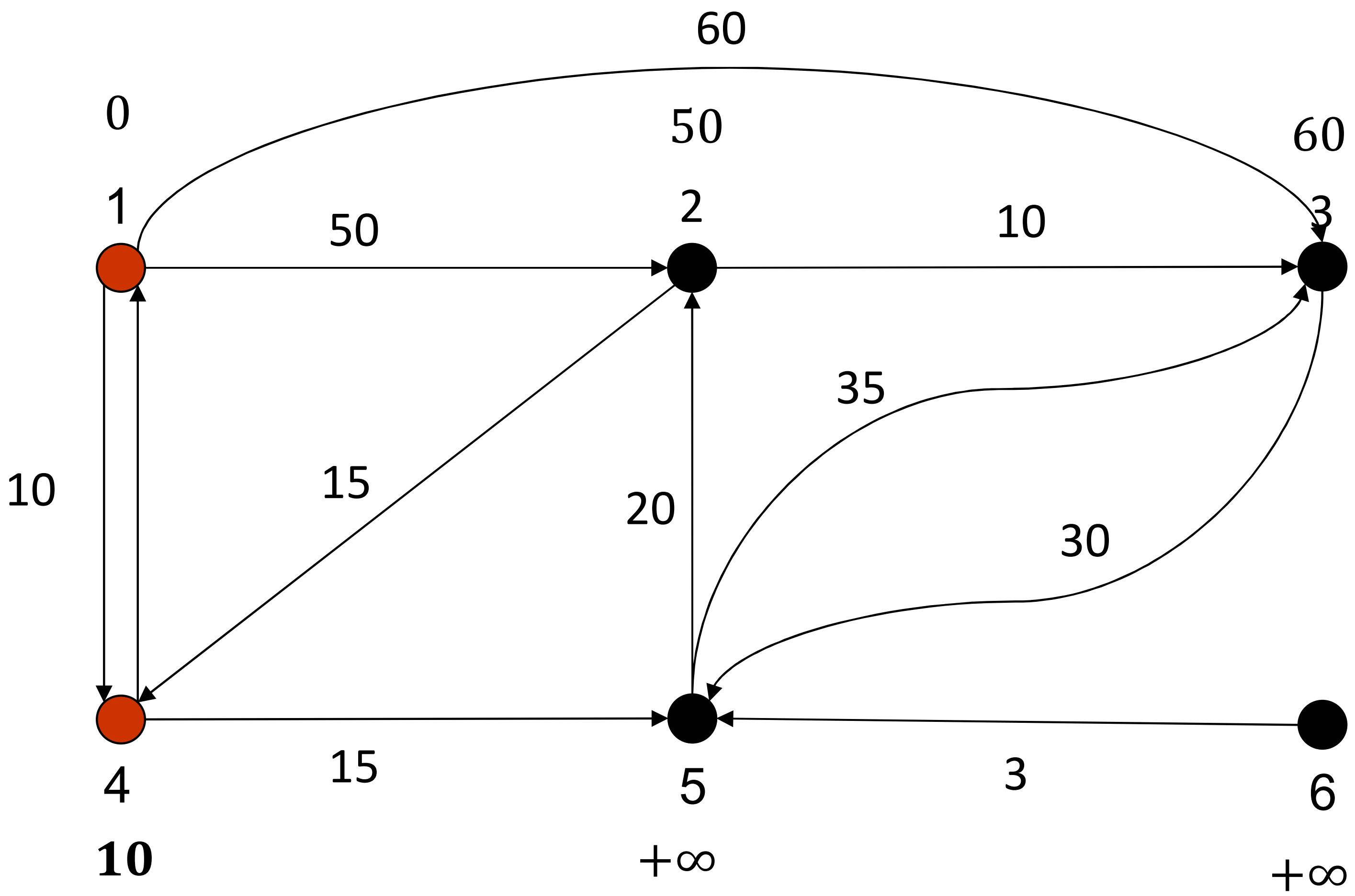
Αρχικοποίηση



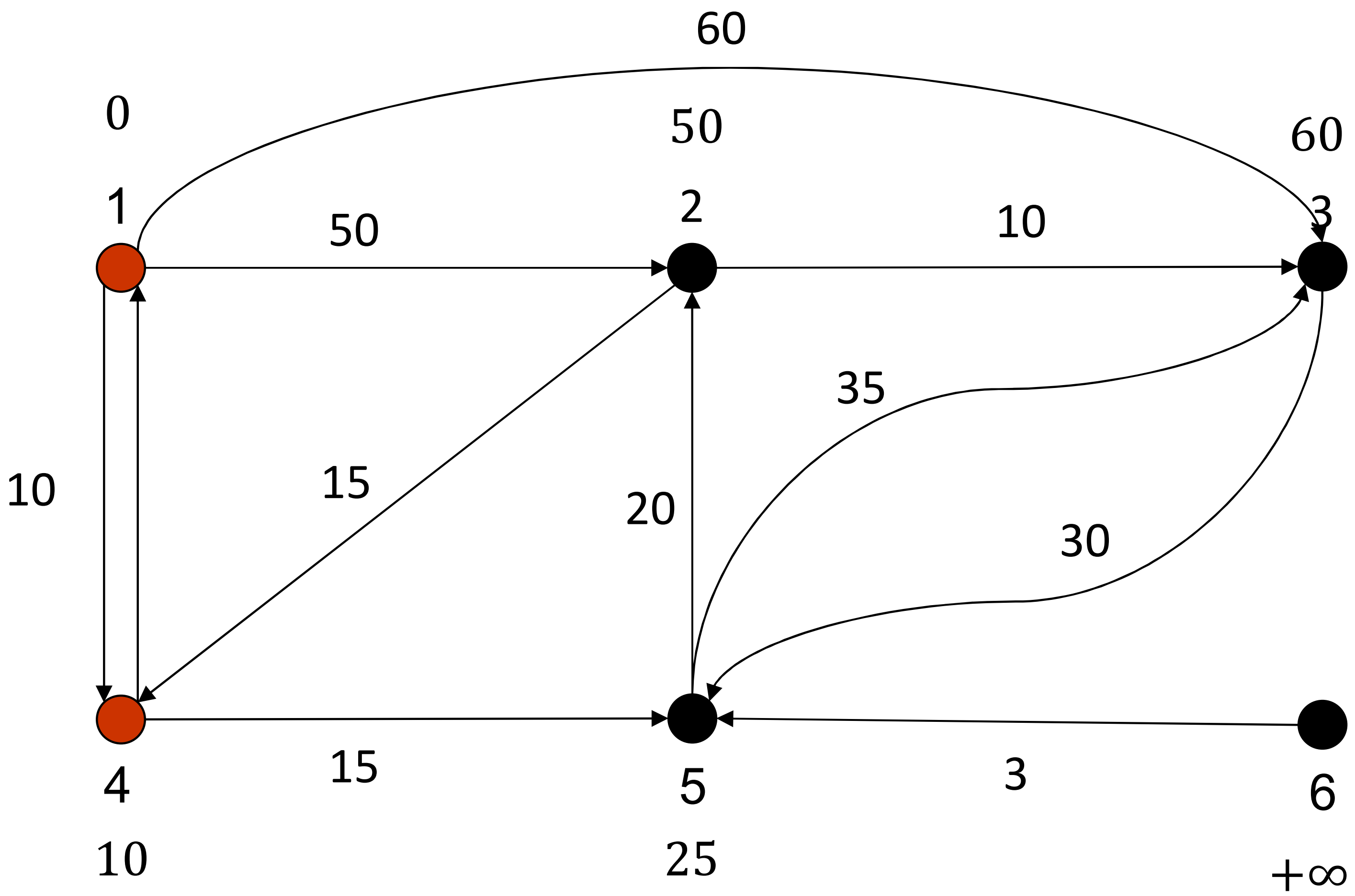
Update όλων των αποστάσεων από του κόμβους που έχουμε επισκεφθεί



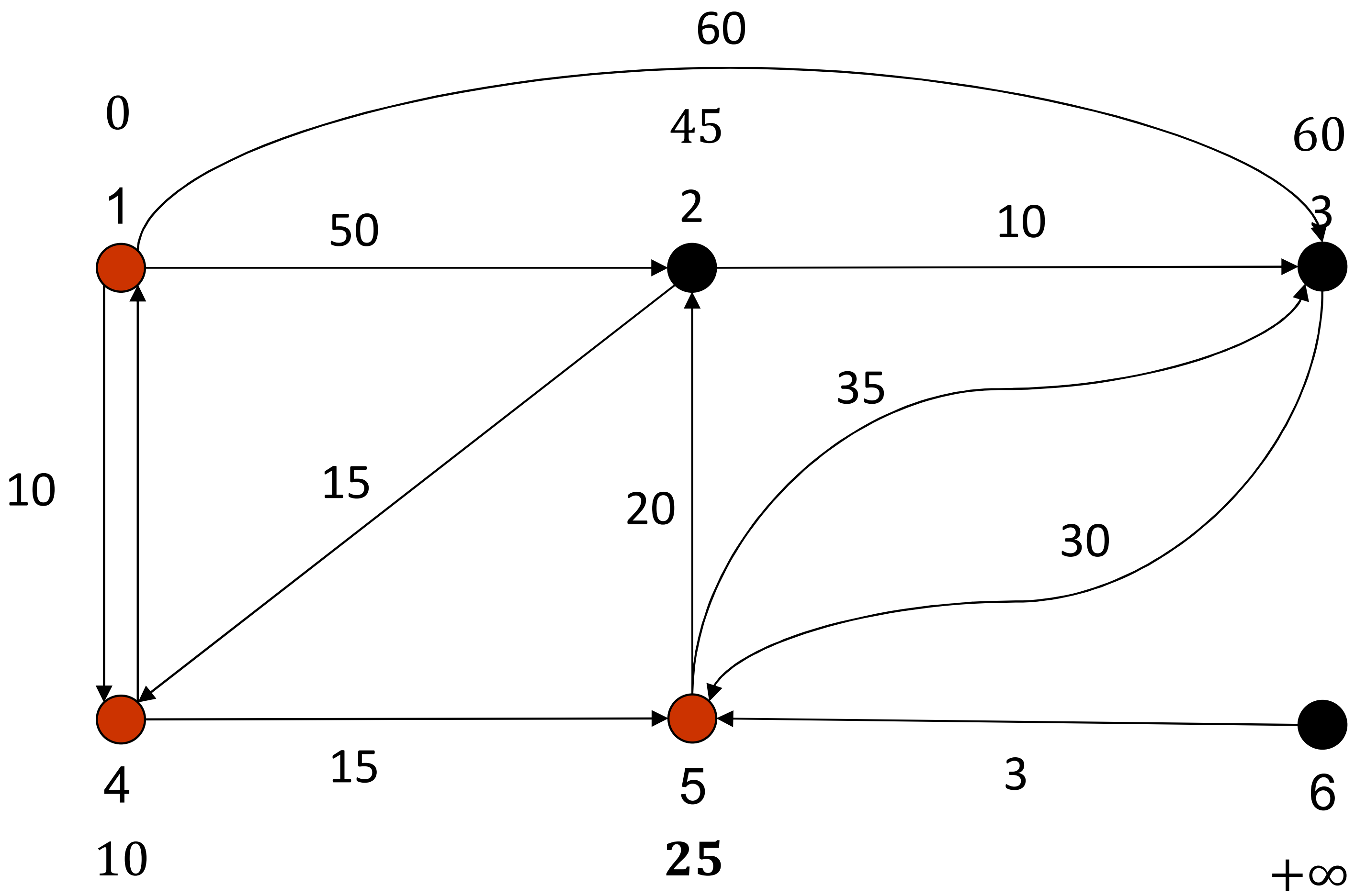
Επιλογή Μικρότερης Απόστασης



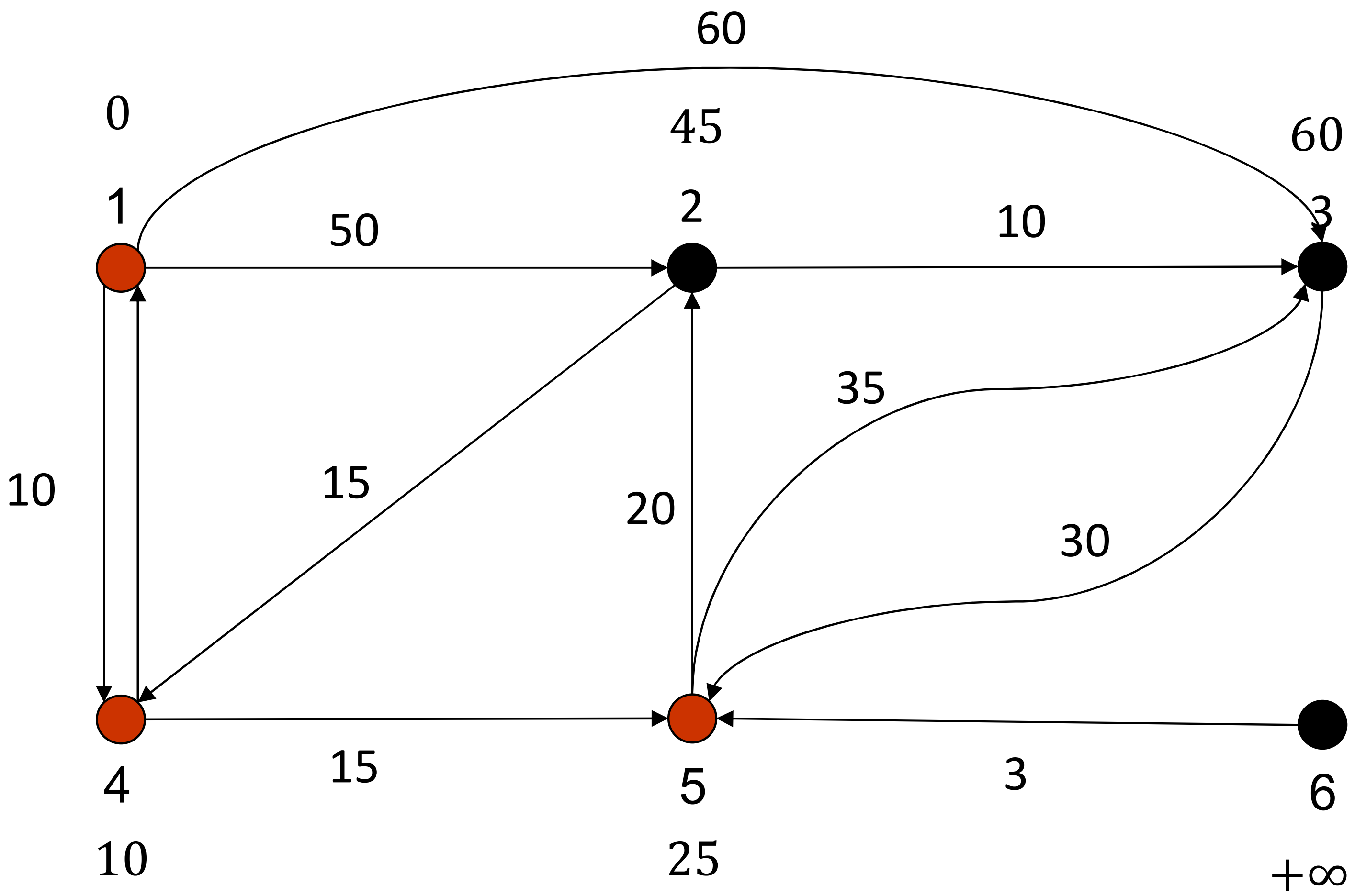
Update όλων των αποστάσεων από του κόμβους που έχουμε επισκεφθεί



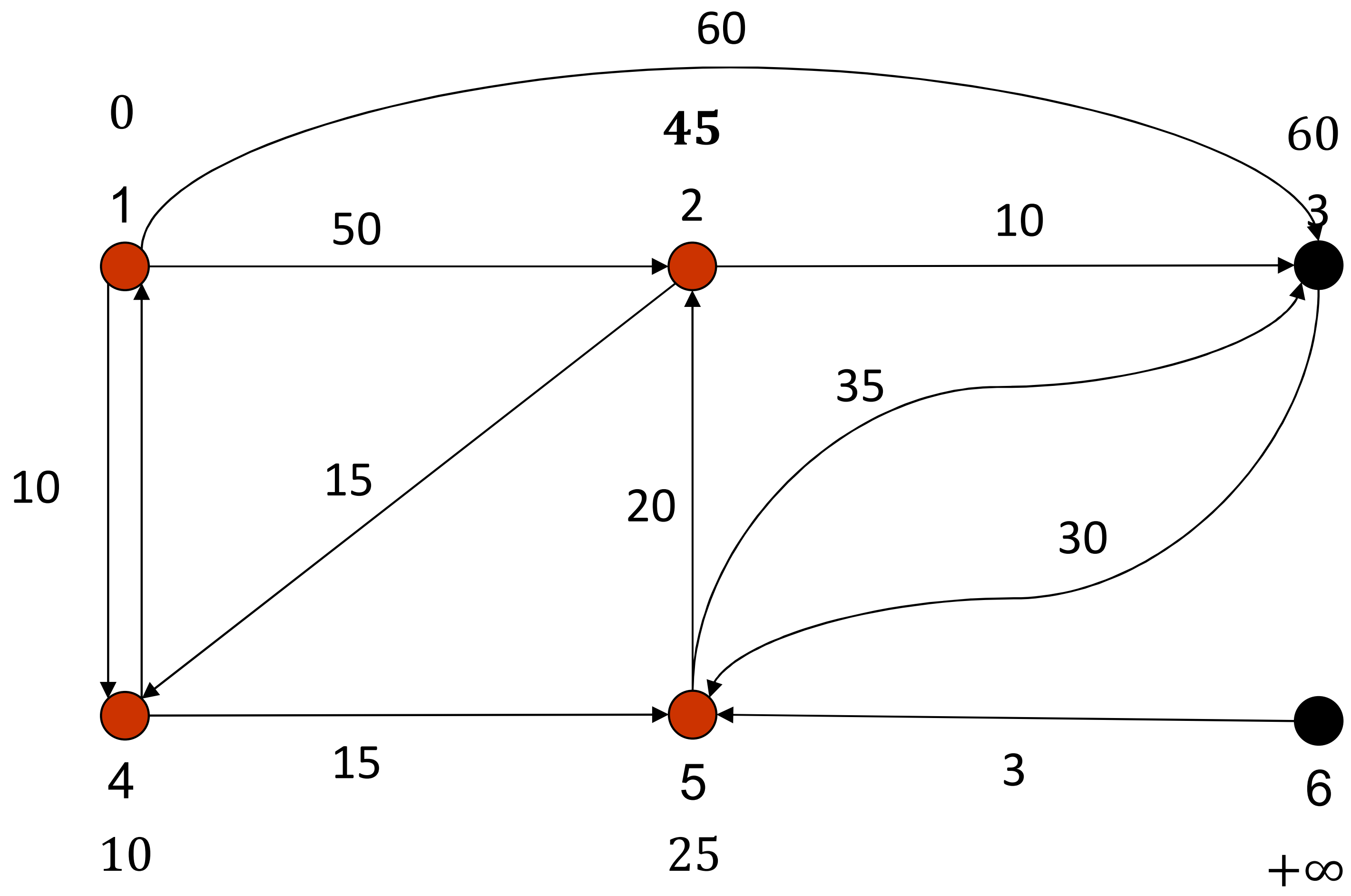
Επιλογή Μικρότερης Απόστασης



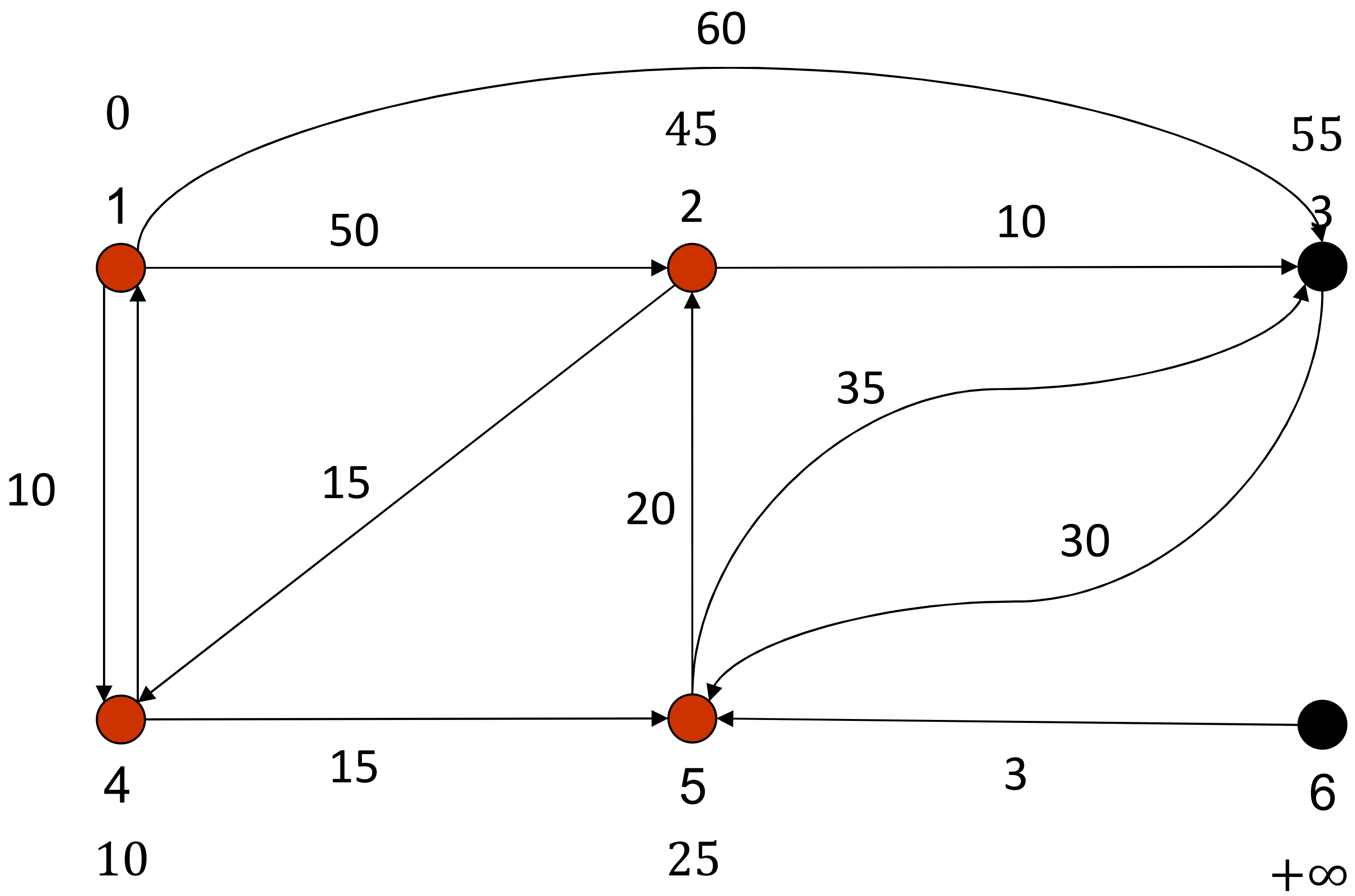
Update όλων των αποστάσεων από του κόμβους που έχουμε επισκεφθεί



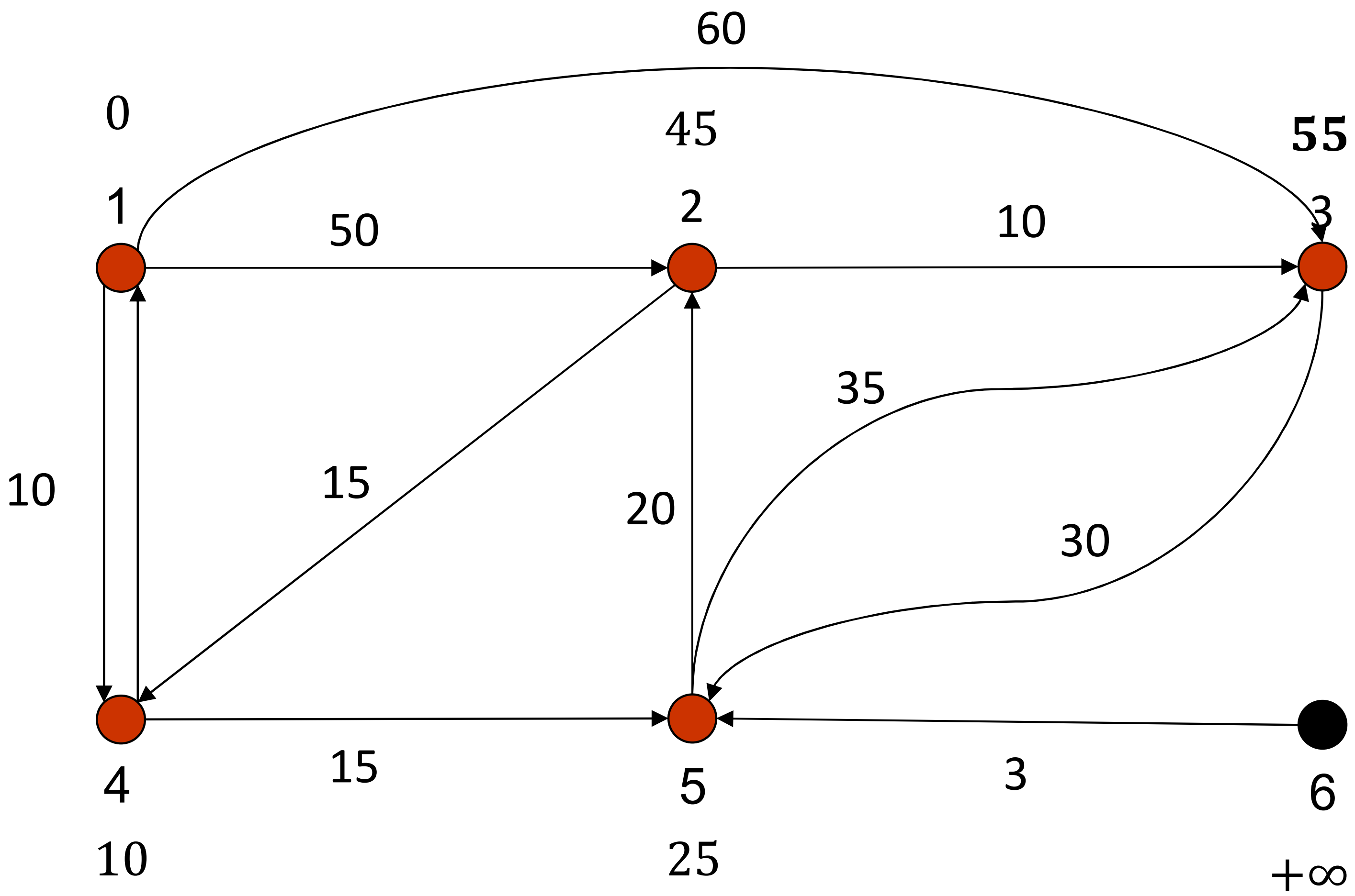
Επιλογή Μικρότερης Απόστασης



Update όλων των αποστάσεων από του κόμβους που έχουμε επισκεφθεί



Update όλων των αποστάσεων από του κόμβους που έχουμε επισκεφθεί



Τερματισμός Αλγορίθμου: Συντομότερο Μονοπάτι το 1-4-5-2-3 με κόστος 55

ΠΡΟΒΛΗΜΑ

- Ας σκεφτούμε ένα άλλο παράδειγμα
- Σε ένα εργοστάσιο, βρίσκονται πέντε μηχανές
- Πρέπει να εγκαταστήσουμε υδραυλικό σύστημα το οποίο να παρέχει την απαραίτητη πίεση και στις πέντε μηχανές.
- Επομένως, πρέπει να ενώσουμε όλες τις μηχανές, με σωληνώσεις
- Δεν είναι απαραίτητο, να είναι ενωμένο κάθε ζεύγος μηχανών
- Είναι όμως απαραίτητο από οποιαδήποτε σε οποιαδήποτε μηχανή να υπάρχει σωλήνωση (η οποία μπορεί να επισκέπτεται και άλλες μηχανές)

ΠΡΟΒΛΗΜΑ Minimum Spanning Tree

- Το πρόβλημα αυτό είναι γνωστό ως minimum spanning tree
- Το Minimum Spanning Tree ορίζεται σε ένα μη κατευθυνόμενο Γράφο.
- Ο γράφος αποτελείται από ένα σύνολο κόμβων και ένα σύνολο εν δυνάμει ακμών
- Στόχος που τίθεται είναι η επιλογή των ακμών έτσι ώστε
  - Υπάρχει ένα μονοπάτι ανάμεσα σε οποιοδήποτε ζεύγος κόμβων
  - Το συνολικό βάρος των επιλεγμένων ακμών να είναι ελάχιστο

ΒΑΣΙΚΗ ΟΡΟΛΟΓΙΑ ΓΙΑ ΜΗ ΚΑΤΕΥΘΥΝΟΜΕΝΟΥΣ ΓΡΑΦΟΥΣ

- Μη κατευθυνόμενος κόμβος
  - Ακμές δεν έχουν φορά
- Μονοπάτι ανάμεσα σε δύο κόμβους
  - Ακολουθία διαφορετικών κόμβων (ή ακμών) η οποία συνδέει τους δύο κόμβους
- Ένα μονοπάτι το οποίο ξεκινά και καταλήγει στο ίδιο σημείο ονομάζεται κύκλος
- Δύο κόμβοι ονομάζονται συνεκτικοί (connected), όταν υπάρχει ένα μονοπάτι (μη κατευθυνόμενοι Γράφοι!) τα οποίο τους συνδέει
- Ένας μη κατευθυνόμενος γράφος ονομάζεται συνεκτικός, κάθε ζεύγος κόμβων του είναι συνεκτικό
- Δέντρο ονομάζεται ένας μη κατευθυνόμενος γράφος κάθε ζεύγος κόμβων του οποίου είναι ενωμένος με ακριβώς ένα μονοπάτι
  - Ισοδύναμα είναι ένα συνεκτικός μη-κατευθυνόμενος γράφος ο οποίος δε περιέχει κύκλους

ΒΑΣΙΚΗ ΟΡΟΛΟΓΙΑ ΓΙΑ ΜΗ ΚΑΤΕΥΘΥΝΟΜΕΝΟΥΣ ΓΡΑΦΟΥΣ

- Ζευγνύον Δέντρο ενός συνόλου n κόμβων ονομάζεται ένας συνεκτικός μη-κατευθυνόμενος γράφος ο οποίος δε περιέχει κύκλους και περιέχει όλους τους n κόμβους
- Κάθε spanning tree έχει ακριβώς $n - 1$ ακμές
  - Ελάχιστος αριθμός τόξων για να έχουμε ένα συνεκτικό γράφο
  - Μέγιστος αριθμός τόξων για να μην έχουμε κύκλους
- Στην εφαρμογή μας ψάχνουμε το spanning tree με μία ακόμα ιδιότητα:

Ελαχιστοποίηση του συνολικού κόστους των επιλεγμένων ακμών

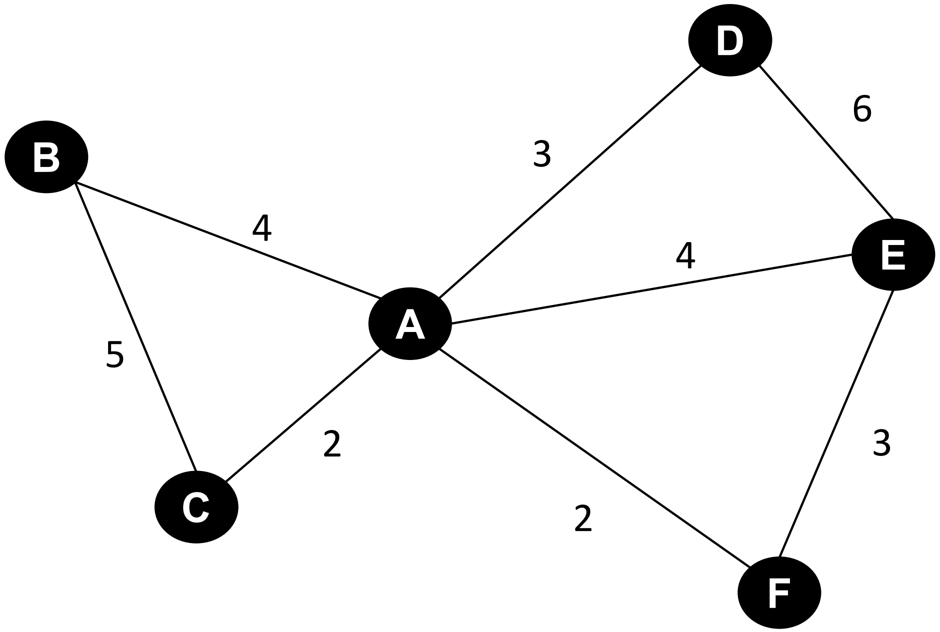
ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΤΟ Minimum Spanning Tree

- Για το Minimum Spanning tree, υπάρχουν δύο ιδιαίτερα γνωστοί πλεονεκτικοί αλγόριθμοι, οι οποίοι εξασφαλίζουν τη βέλτιστη λύση
- Εξασφαλίζουν δηλαδή πως το δέντρο τοι οποίο θα παραχθεί είναι το minimum spanning tree
  - Prims Algorithm
  - Kruskall Algorithm

Prims Algorithm

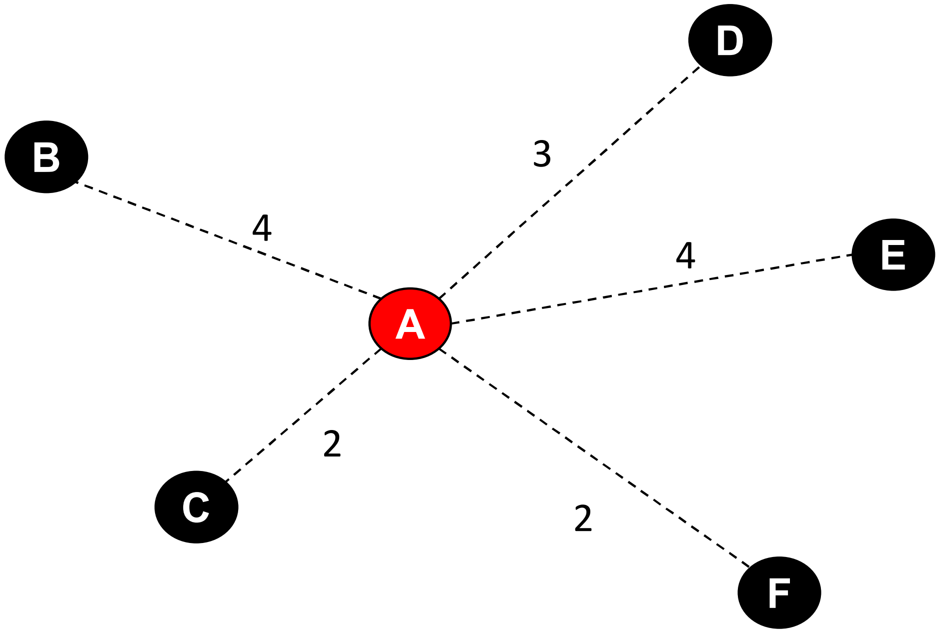
- Ξεκινώντας από ένα συνδεδεμένο γράφο με n κόμβους και δεδομένα βάρη, ονόμασε τις ακμές εν δυνάμει ακμές
- Διάλεξε τυχαία ένα κόμβο (A)
- Βρες τον κοντινότερο του κόμβο (B)
- $E_S \leftarrow E_S \cup \{(A, B)\}$
- $N_C \leftarrow N_C \cup \{A\} \cup \{B\}$
- Το σύνολο των επιλεγμένω ακμών $E_S = \emptyset$
- While ($|E_S| < n - 1$)
  - Βρες τι μικρότερη ακμή η οποία ενώνει ένα κόμβο του N_C (A) με ένα κόμβο του $N - N_C$ (B)
  - $E_S \leftarrow E_S \cup \{A, B\}$
  - $N_C \leftarrow N_C \cup \{B\}$

Prims Algorithm



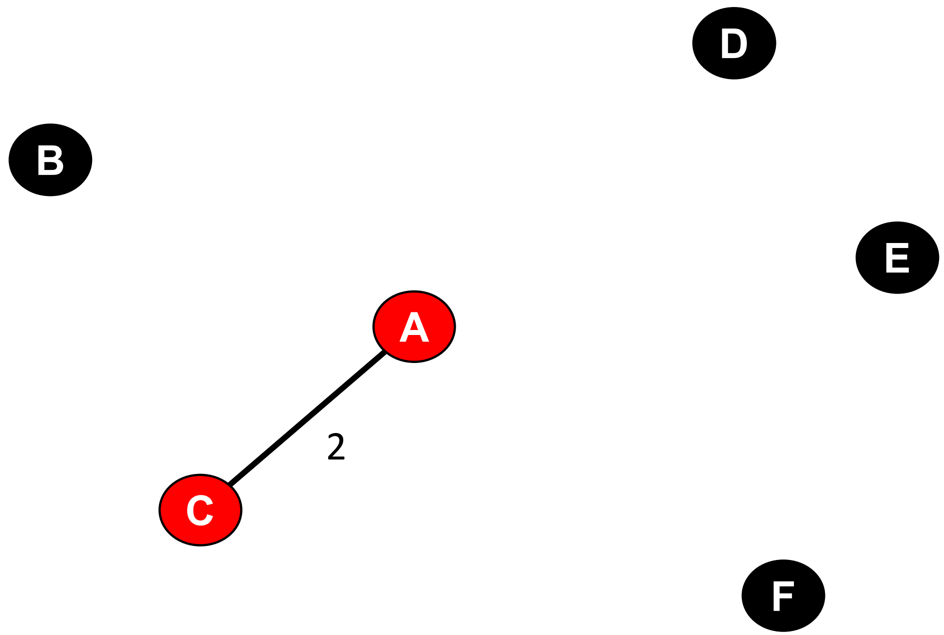
Prims Algorithm

- Τυχαία Επιλογή Κόμβου: A



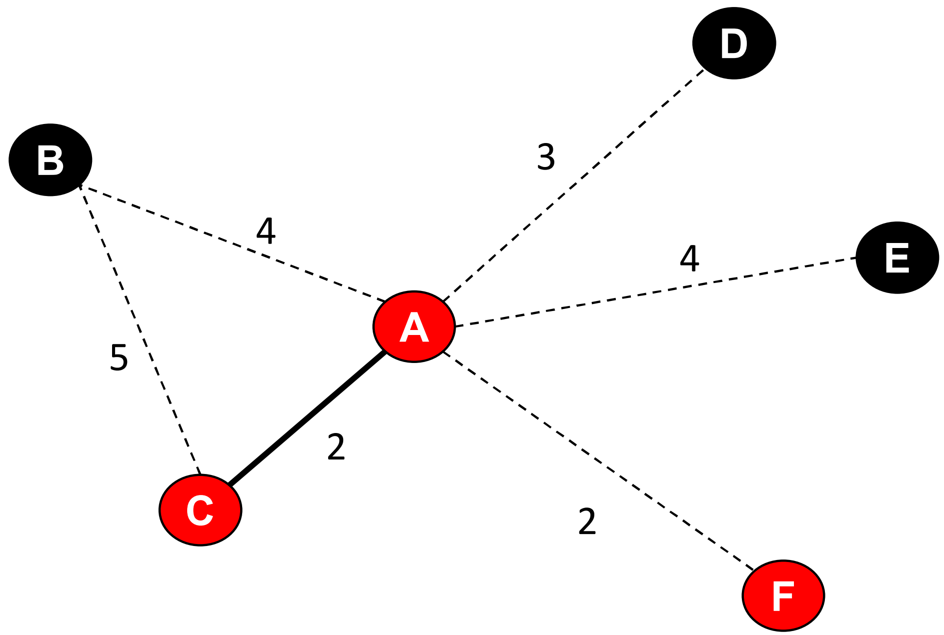
Prims Algorithm

- Επιλογή της οικονομικότερης σύνδεσης {A, C}
- $E_S = \{(A, \mathrm{C})\}$
- $N_C = \{A, C\}$



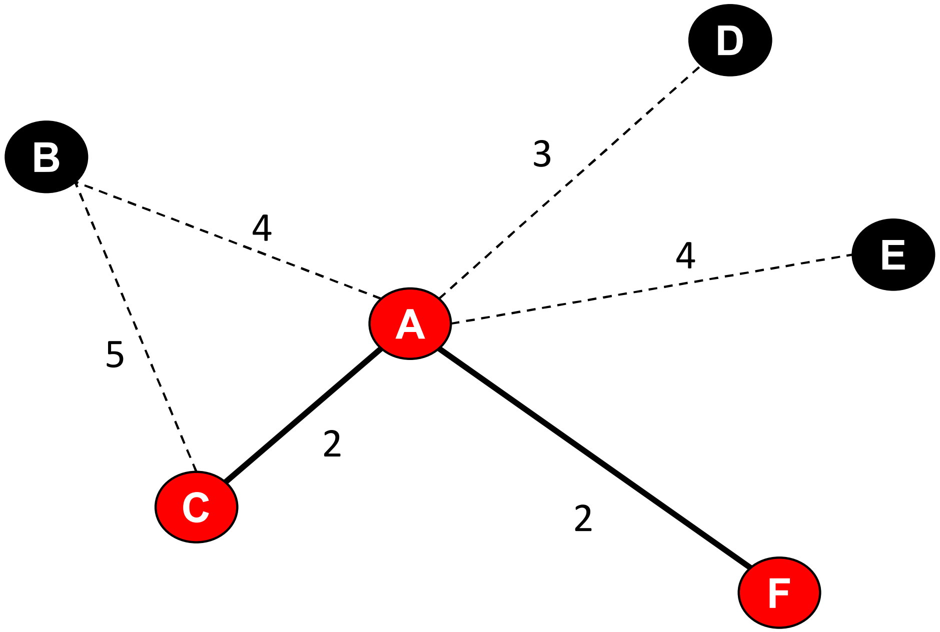
Prims Algorithm

- Εξέταση όλων των ακμών από τους κόμβους του N_C προς τους κόμβους εκτός του N_C



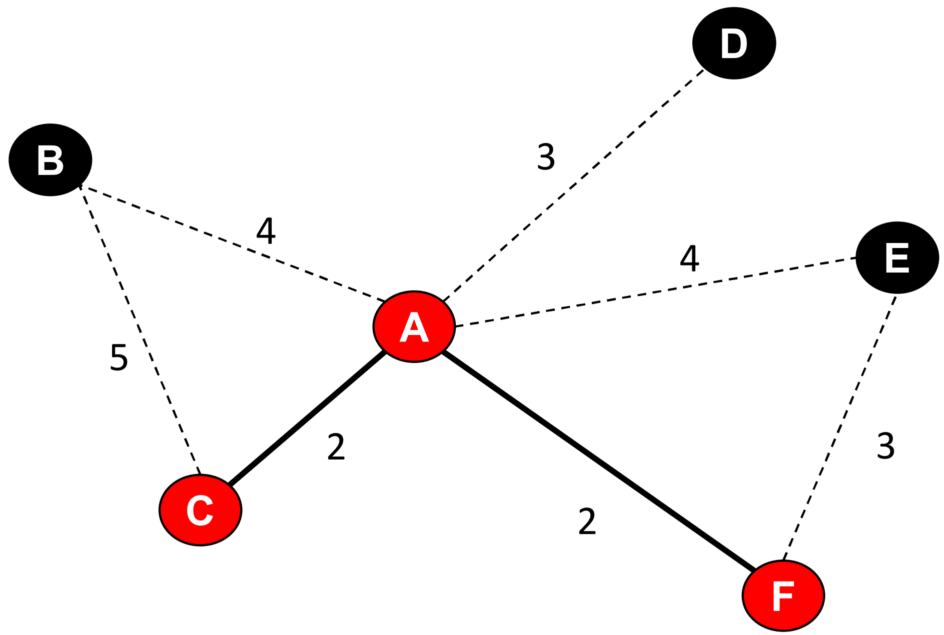
Prims Algorithm

- Επιλογή της οικονομικότερης σύνδεσης {A, C}
- $E_S = \{(A, \mathrm{C}), (A, F)\}$
- $N_C = \{A, C, F\}$



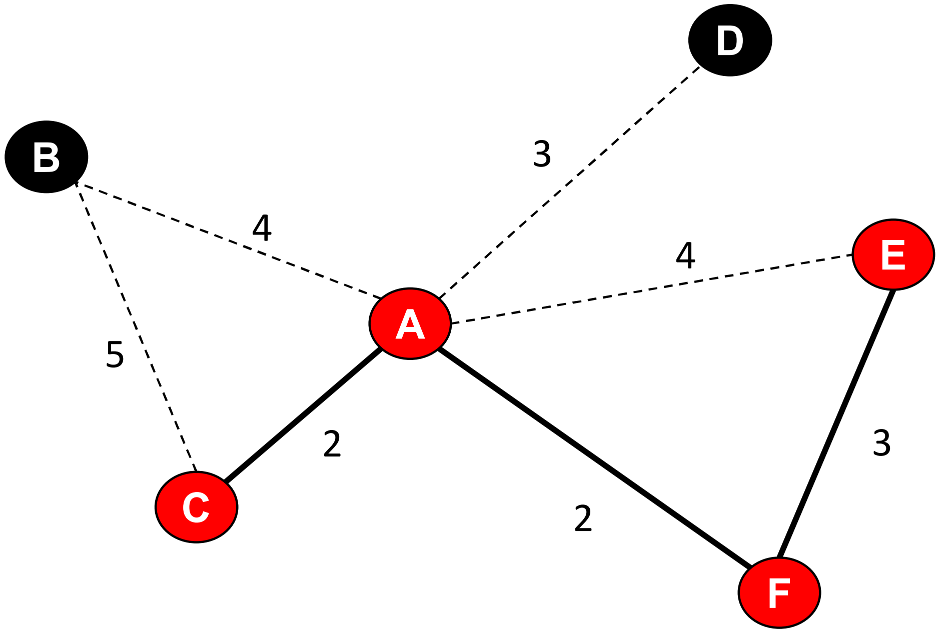
Prims Algorithm

- Εξέταση όλων των ακμών από τους κόμβους του N_C προς τους κόμβους εκτός του N_C



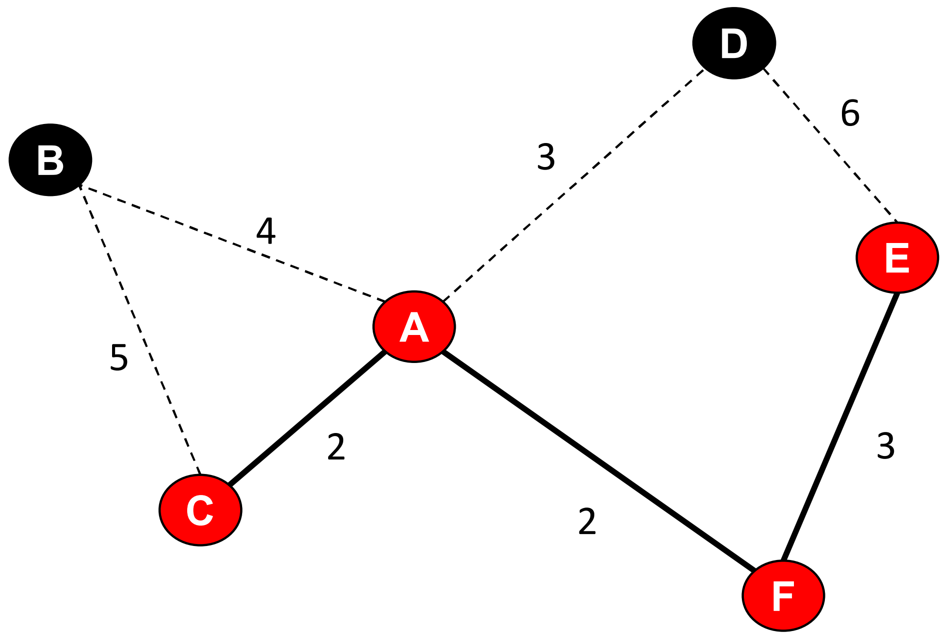
Prims Algorithm

- Επιλογή της οικονομικότερης σύνδεσης {A, C}
- $E_S = \{(A, \mathrm{C}), (A, F), (F, E)\}$
- $N_C = \{A, C, F, E\}$



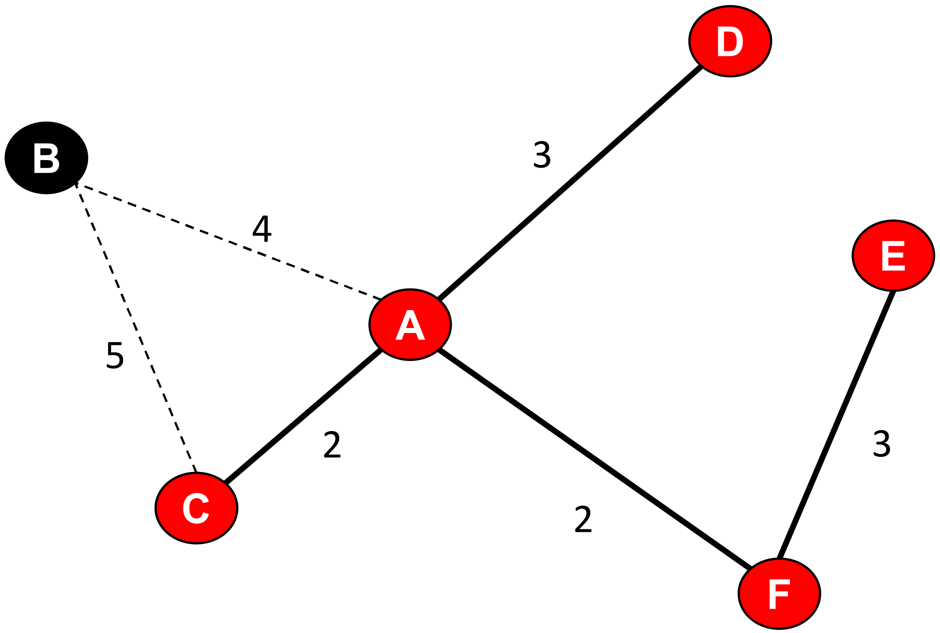
Prims Algorithm

- Εξέταση όλων των ακμών από τους κόμβους του N_C προς τους κόμβους εκτός του N_C



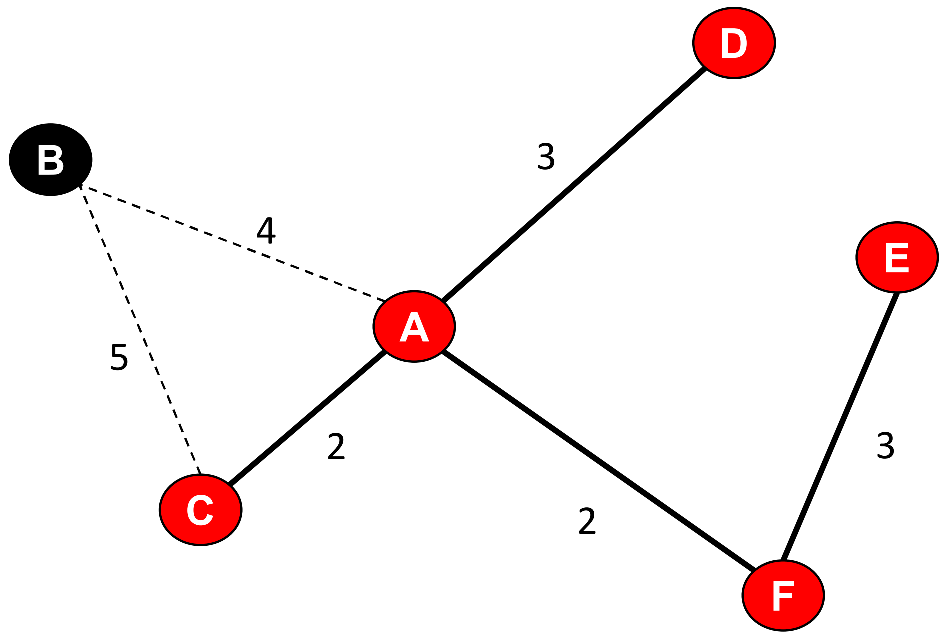
Prims Algorithm

- Επιλογή της οικονομικότερης σύνδεσης {A, C}
- $E_S = \{(A, \mathrm{C}), (A, F), (F, E), (A, D)\}$
- $N_C = \{A, C, F, E, D\}$



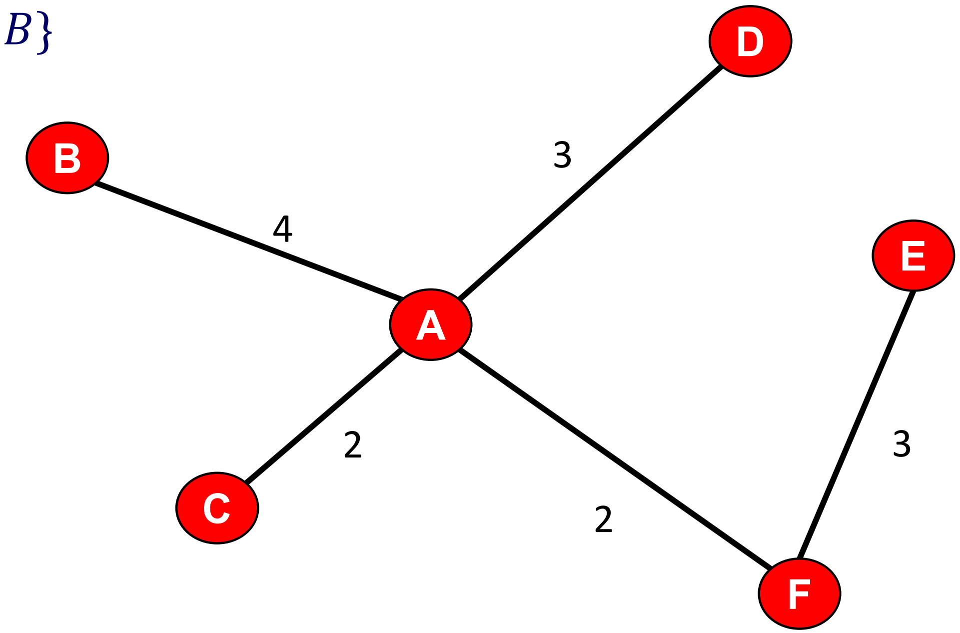
Prims Algorithm

- Εξέταση όλων των ακμών από τους κόμβους του N_C προς τους κόμβους εκτός του N_C



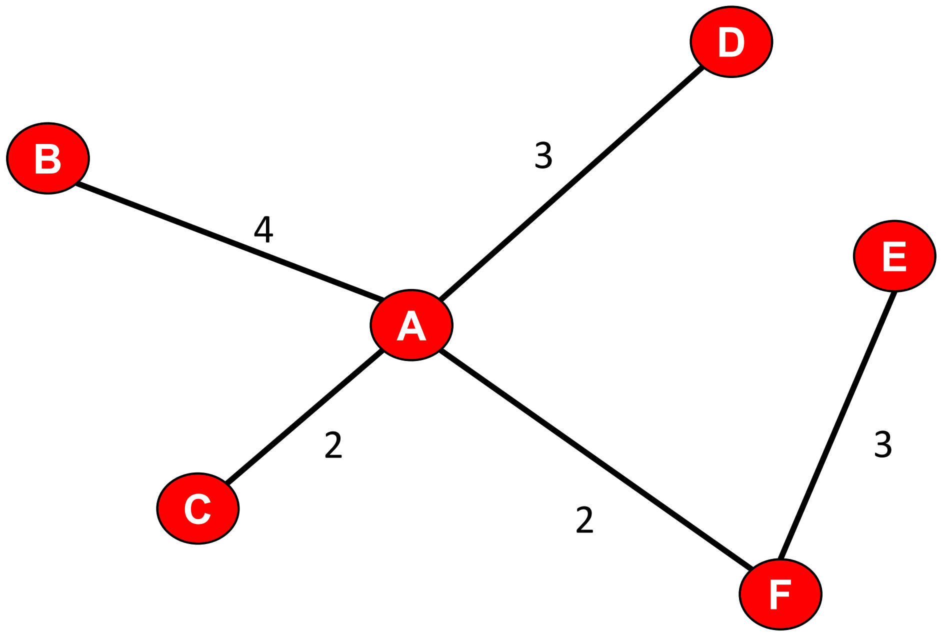
Prims Algorithm

- Επιλογή της οικονομικότερης σύνδεσης {A, C}
- $E_S = \{(A, \mathrm{C}), (A, F), (F, E), (A, D), (A, B)\}$
- $N_C = \{A, C, F, E, D, B\}$



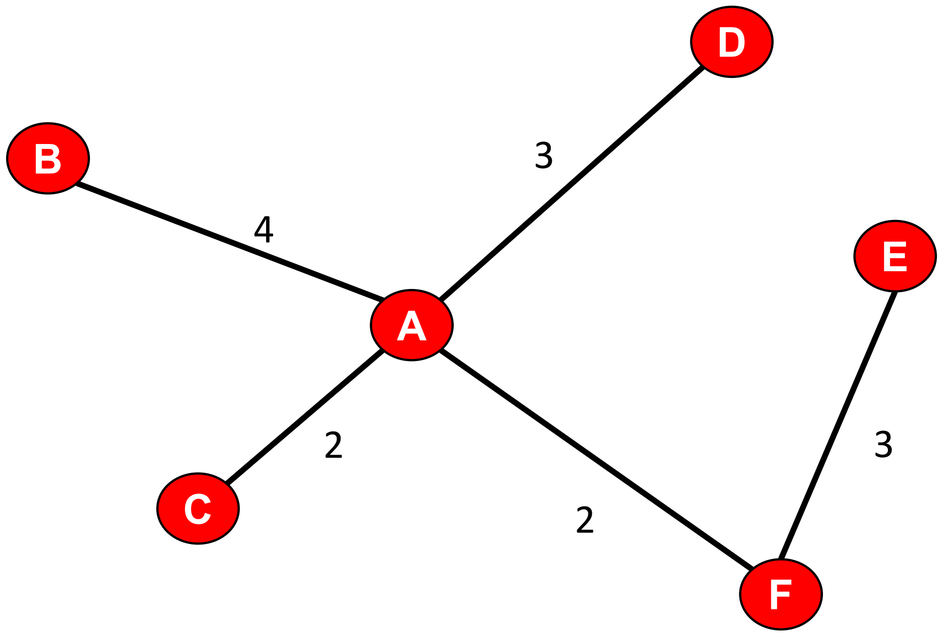
Prims Algorithm

- Τερματισμός Αλγορίθμου
- $E_S = \{(A, \mathrm{C}), (A, F), (F, E), (A, D), (A, B)\}$
- $N_C = \{A, C, F, E, D, B\}$
- *Τελικό Κόστος: 14*



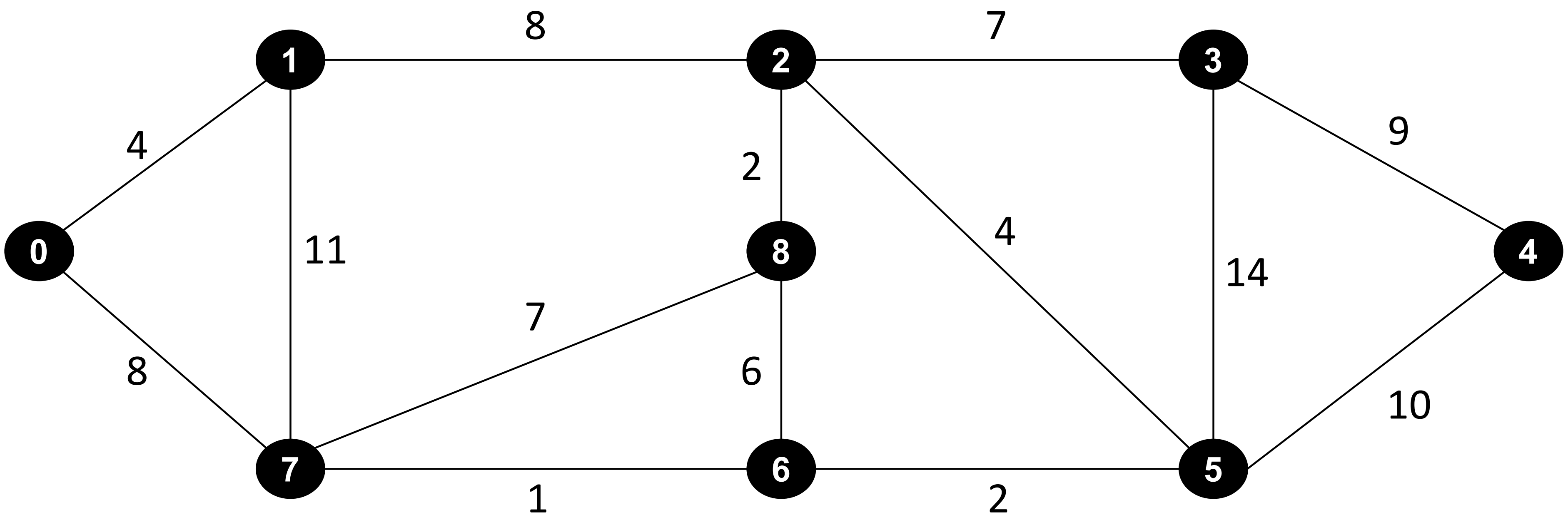
Prims Algorithm

- Τερματισμός Αλγορίθμου
- $E_S = \{(A, \mathrm{C}), (A, F), (F, E), (A, D), (A, B)\}$
- $N_C = \{A, C, F, E, D, B\}$
- *Τελικό Κόστος: 14*



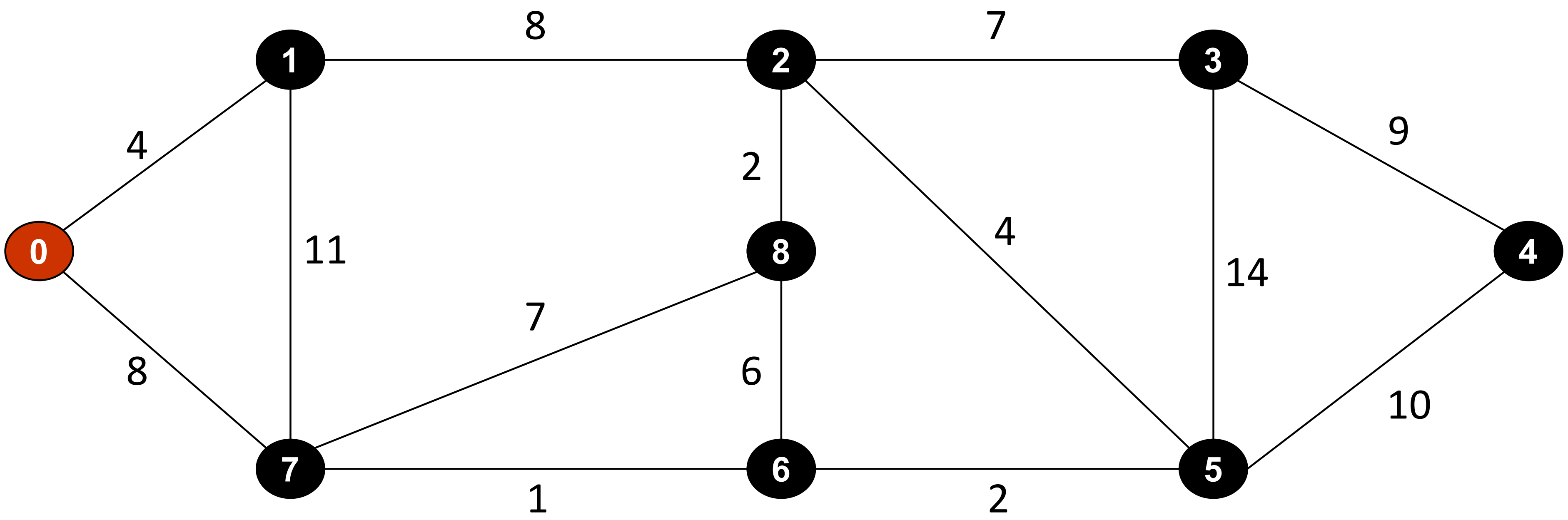
Prims Algorithm

- Εφαρμόστε τον αλγόριθμο Prims στο παρακάτω γράφο, ώστε να σχηματιστεί το Minimum Spanning Tree του συνόλου των 9 κόμβων



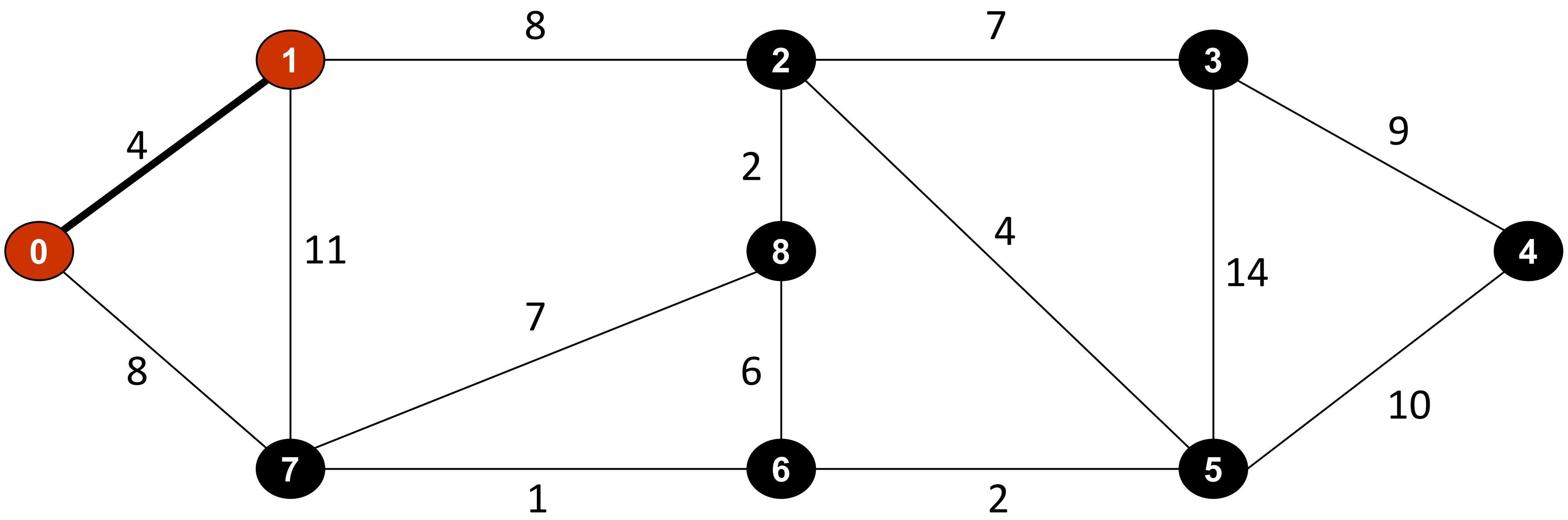
Prims Algorithm

- Αρχικοποίηση
- Έστω η Τυχαία επιλογή του 0



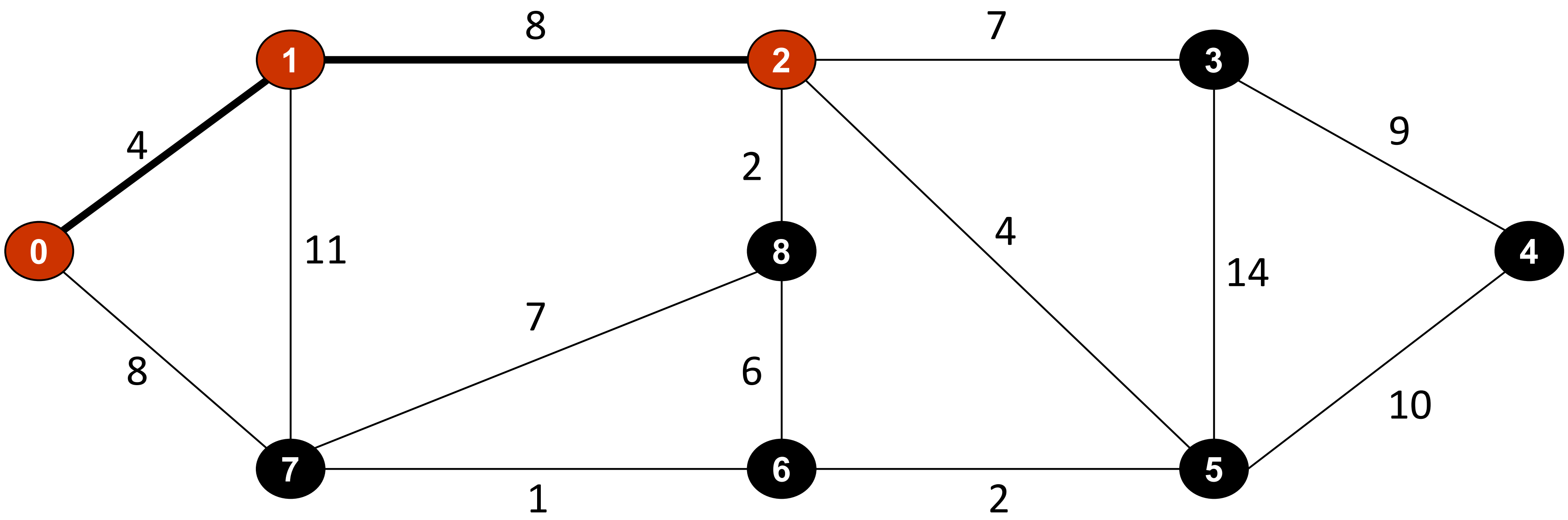
Prims Algorithm

- Επιλογή της σύνδεσης 0-1
  - $N_C \leftarrow \{0\} \cup \{B\}$
  - $E_S \leftarrow \{0,1\}$



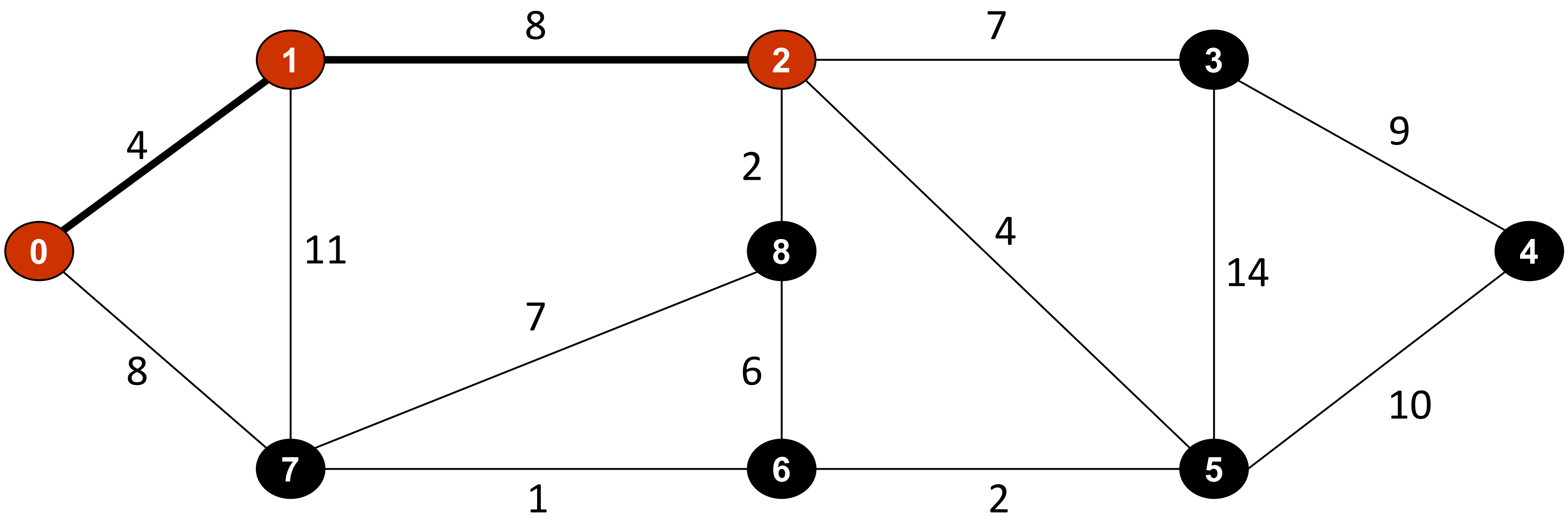
Prims Algorithm

- Επιλογή οικονομικότερης σύνδεσης



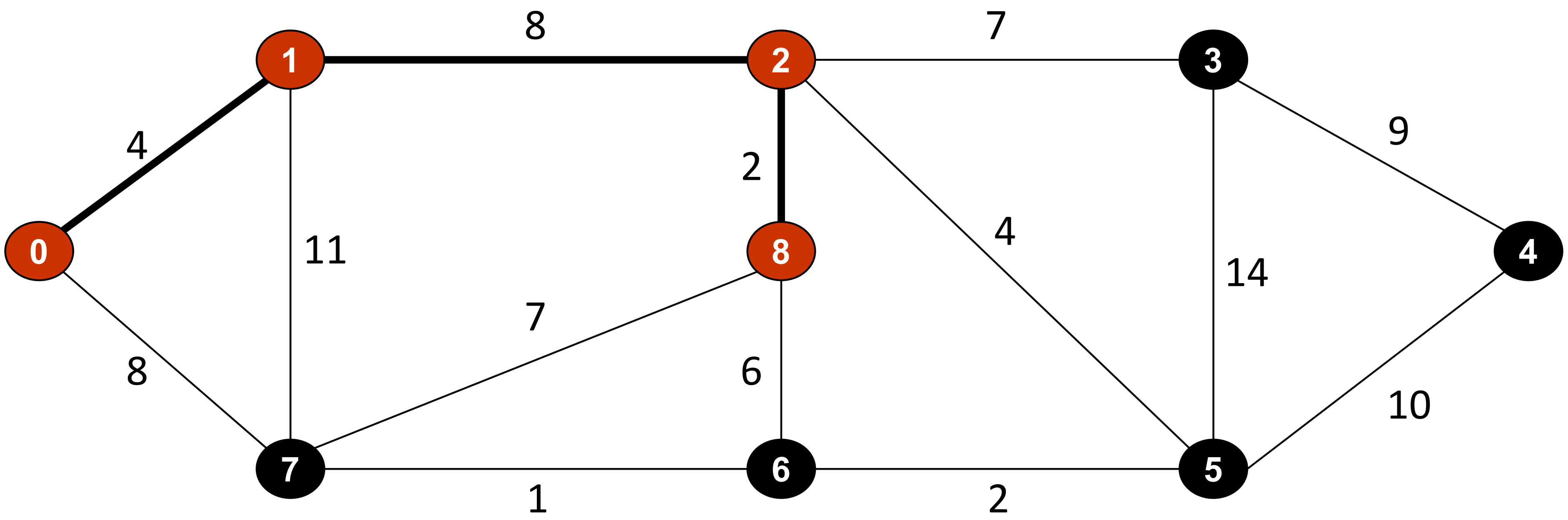
Prims Algorithm

- Επιλογή της σύνδεσης 0-1
  - $N_C \leftarrow \{0,1\} \cup \{2\}$
  - $E_S \leftarrow \{(0,1)\} \cup \{(1,2)\}$



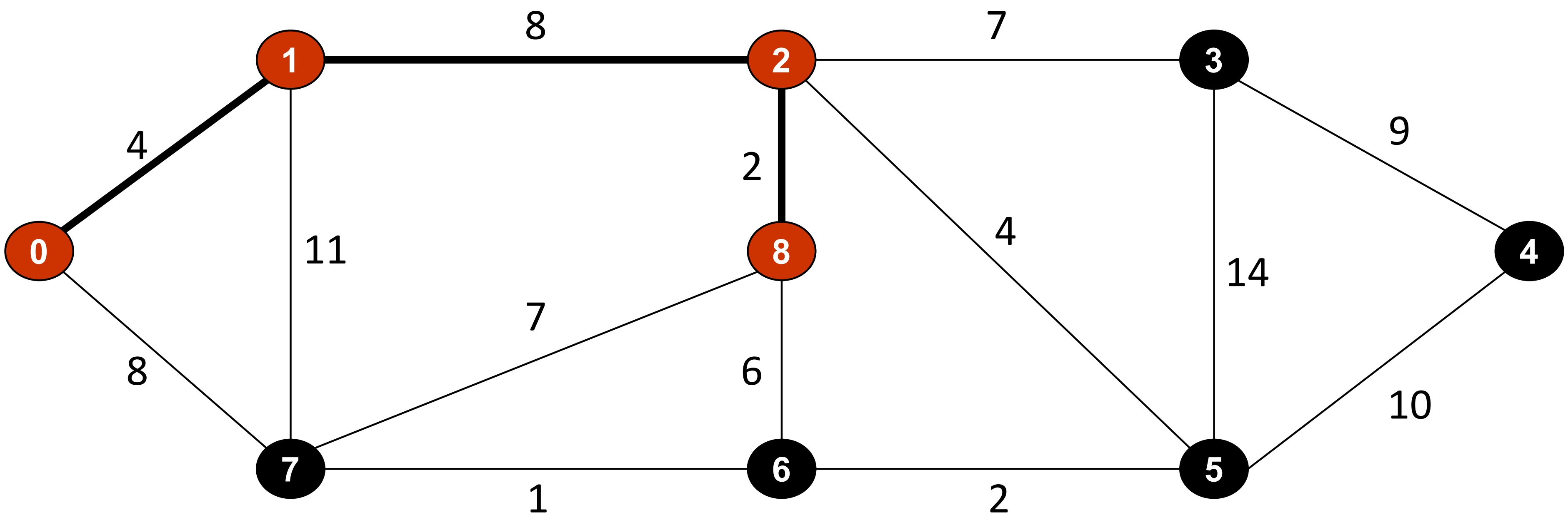
Prims Algorithm

- Επιλογή οικονομικότερης σύνδεσης



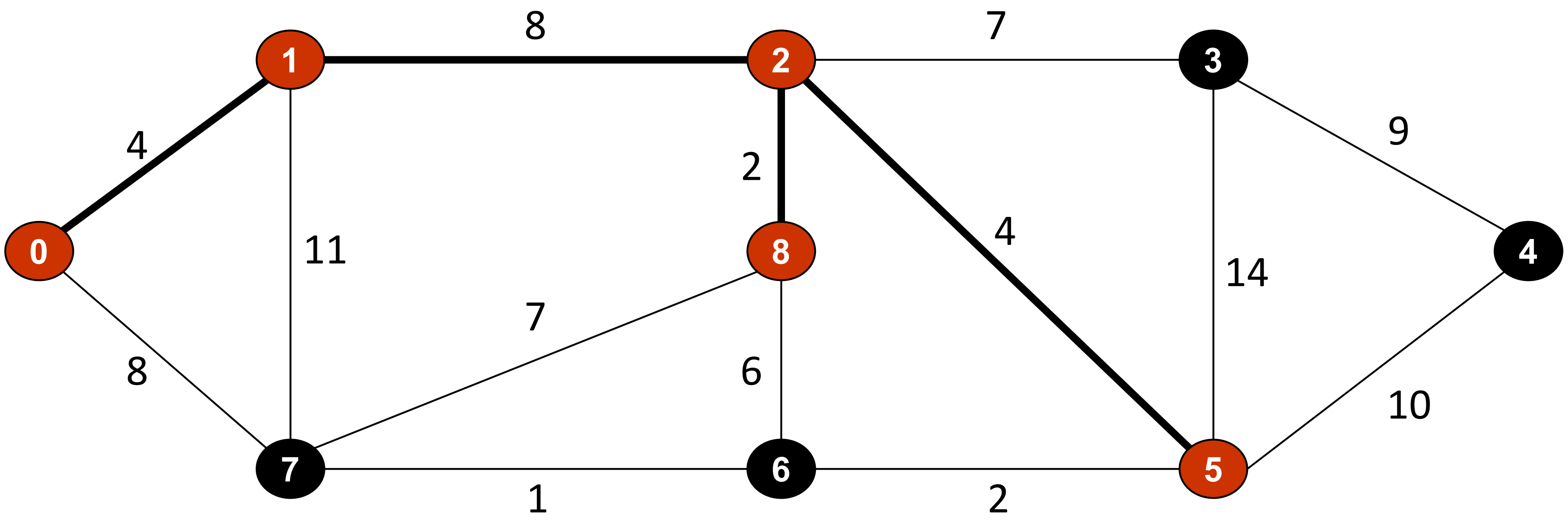
Prims Algorithm

- Επιλογή της σύνδεσης 0-1
  - $N_C \leftarrow \{0{,}1{,}2\} \cup \{8\}$
  - $E_S \leftarrow \{(0{,}1), (1{,}2)\} \cup \{(2{,}8)\}$



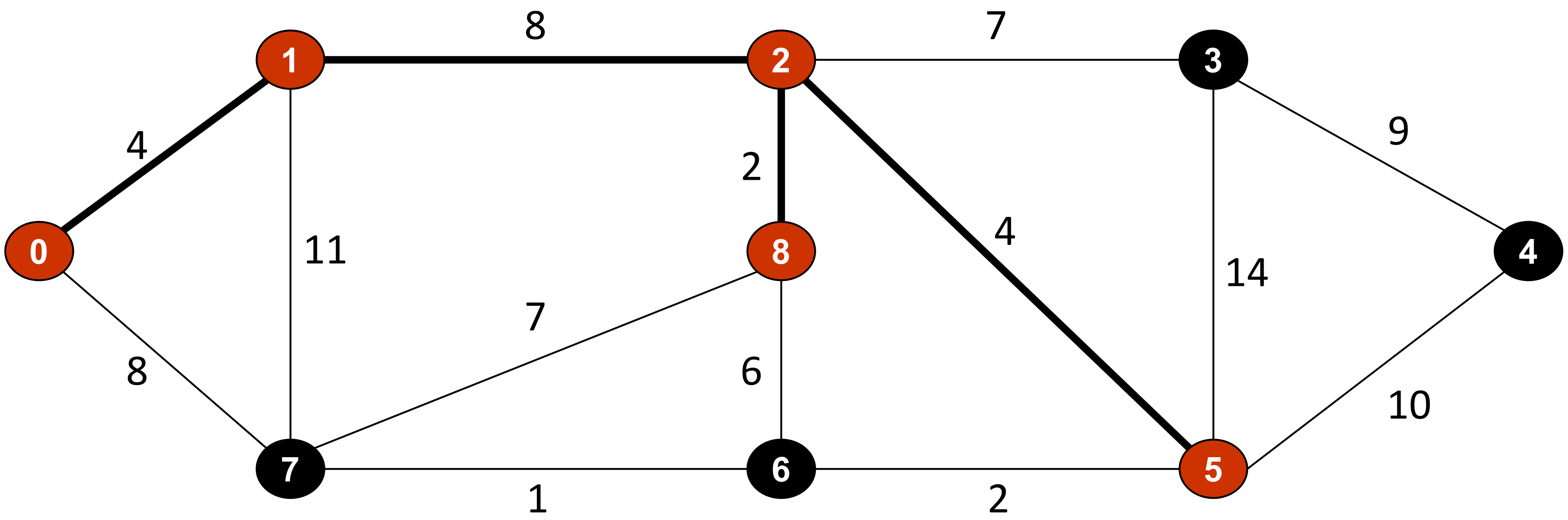
Prims Algorithm

- Επιλογή οικονομικότερης σύνδεσης



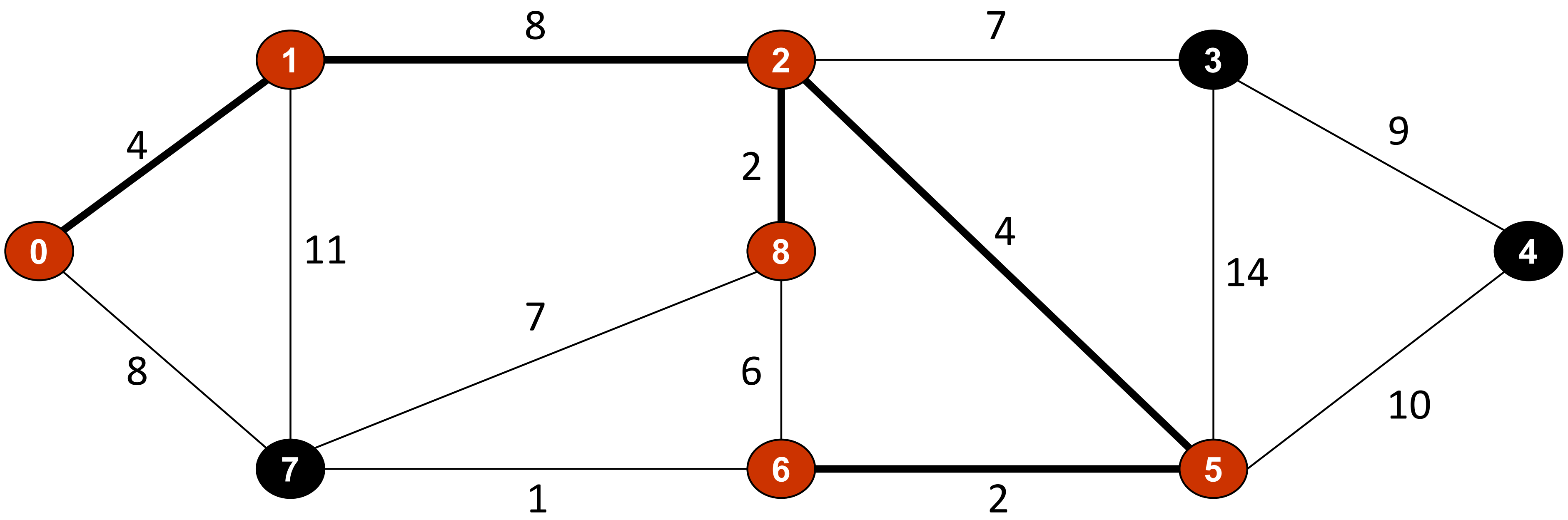
Prims Algorithm

- Επιλογή της σύνδεσης 0-1
  - $N_C \leftarrow \{0,1,2,8\} \cup \{5\}$
  - $E_S \leftarrow \{(0,1), (1,2), (2,8)\} \cup \{(2,5)\}$



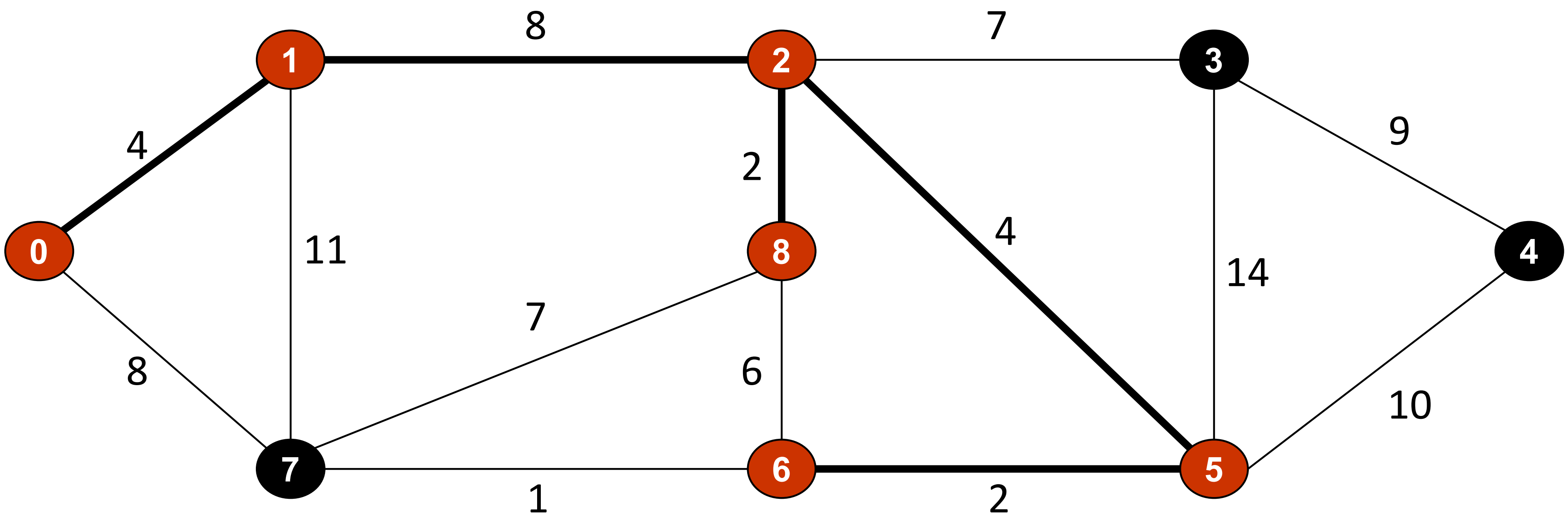
Prims Algorithm

- Επιλογή οικονομικότερης σύνδεσης



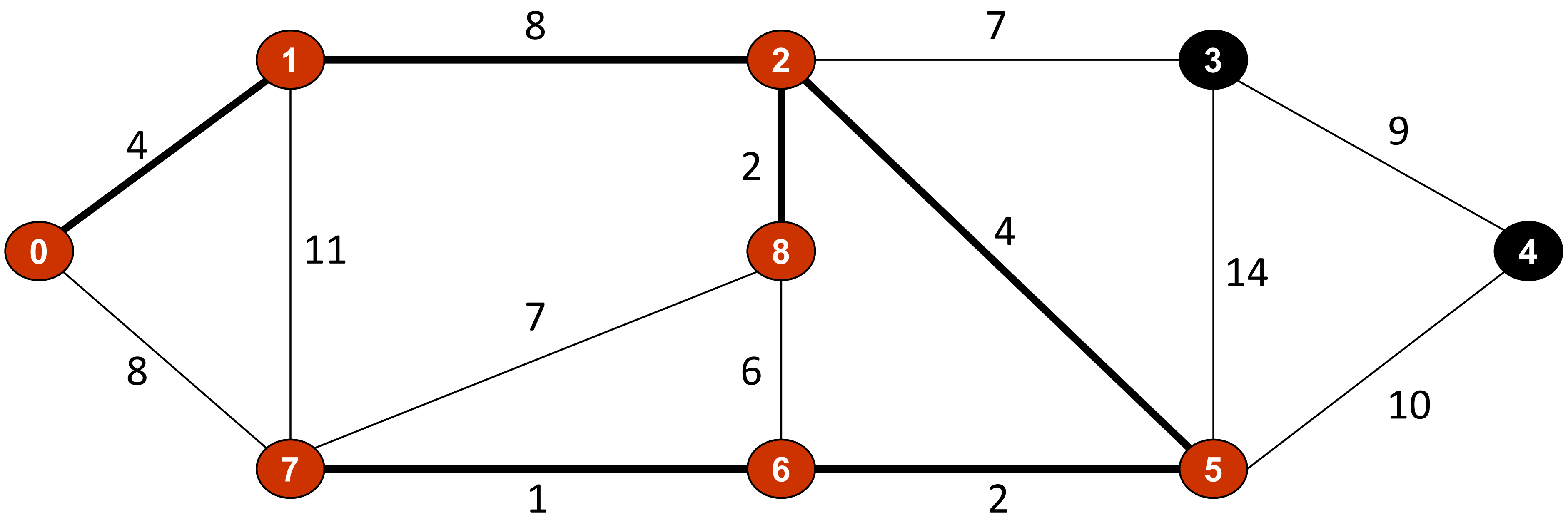
Prims Algorithm

- Επιλογή της σύνδεσης 5-6
  - $N_C \leftarrow \{0,1,2,8,5\} \cup \{5\}$
  - $E_S \leftarrow \{(0,1),(1,2),(2,8),(2,5)\} \cup \{(5,6)\}$



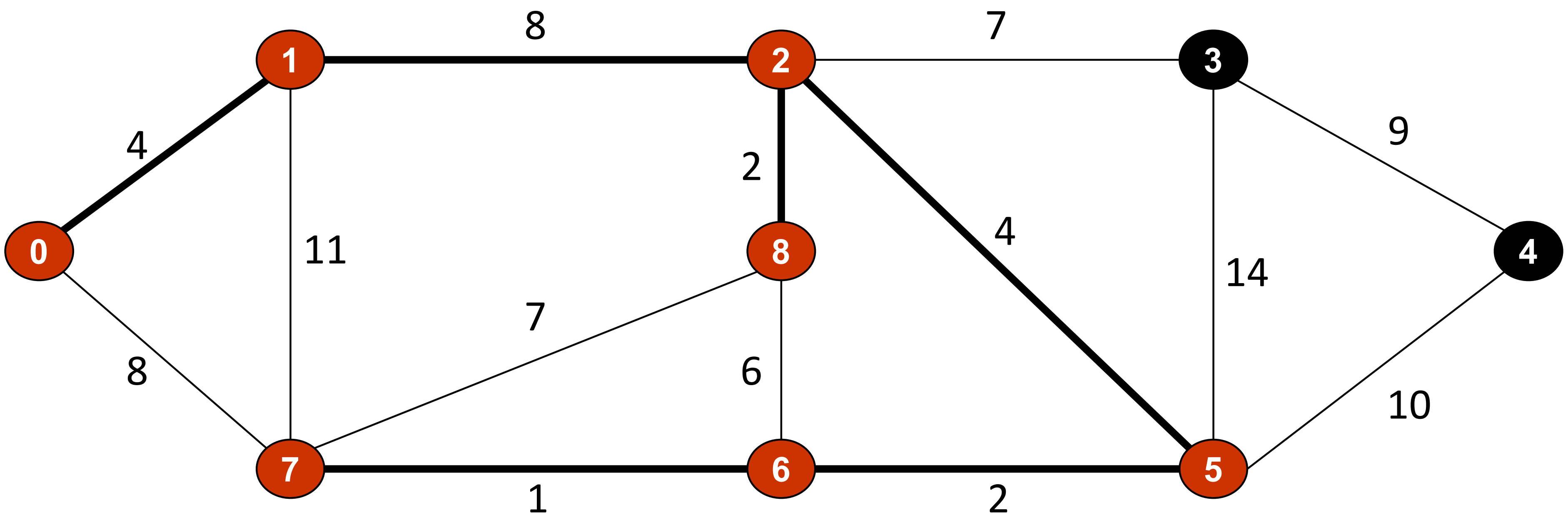
Prims Algorithm

- Επιλογή οικονομικότερης σύνδεσης



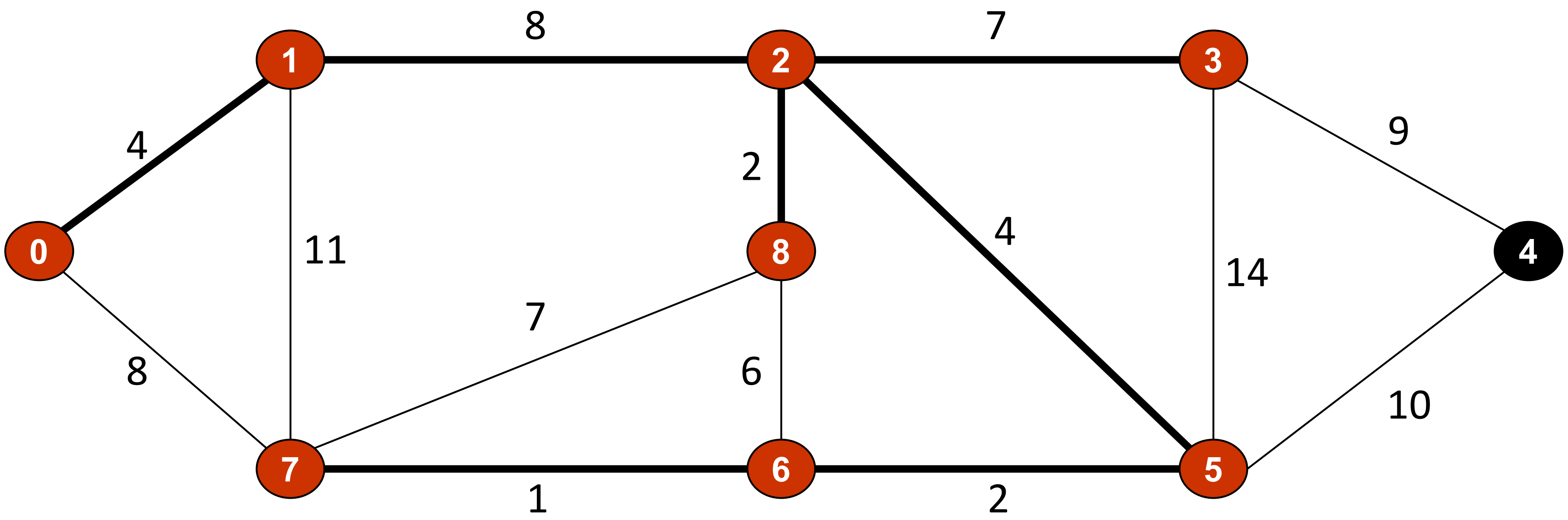
Prims Algorithm

- Επιλογή της σύνδεσης 6-7
  - $N_C \leftarrow \{0,1,2,8,5,6\} \cup \{7\}$
  - $E_S \leftarrow \{(0,1), (1,2), (2,8), (2,5), (5,6)\} \cup \{(6,7)\}$



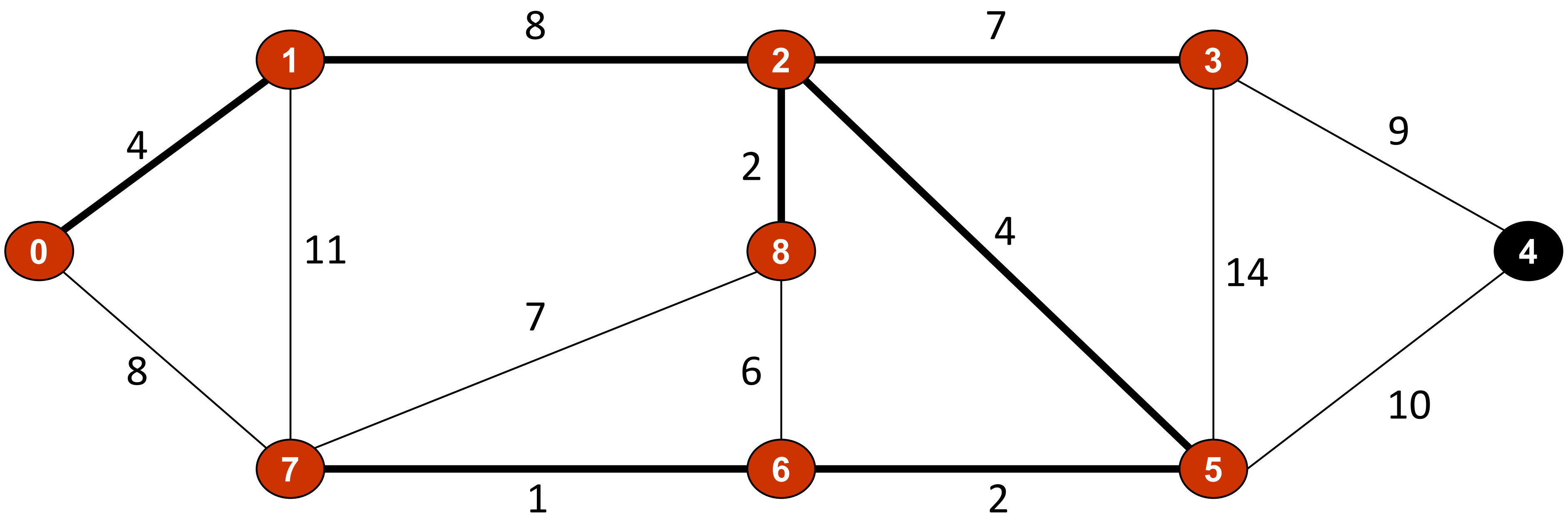
Prims Algorithm

- Επιλογή οικονομικότερης σύνδεσης



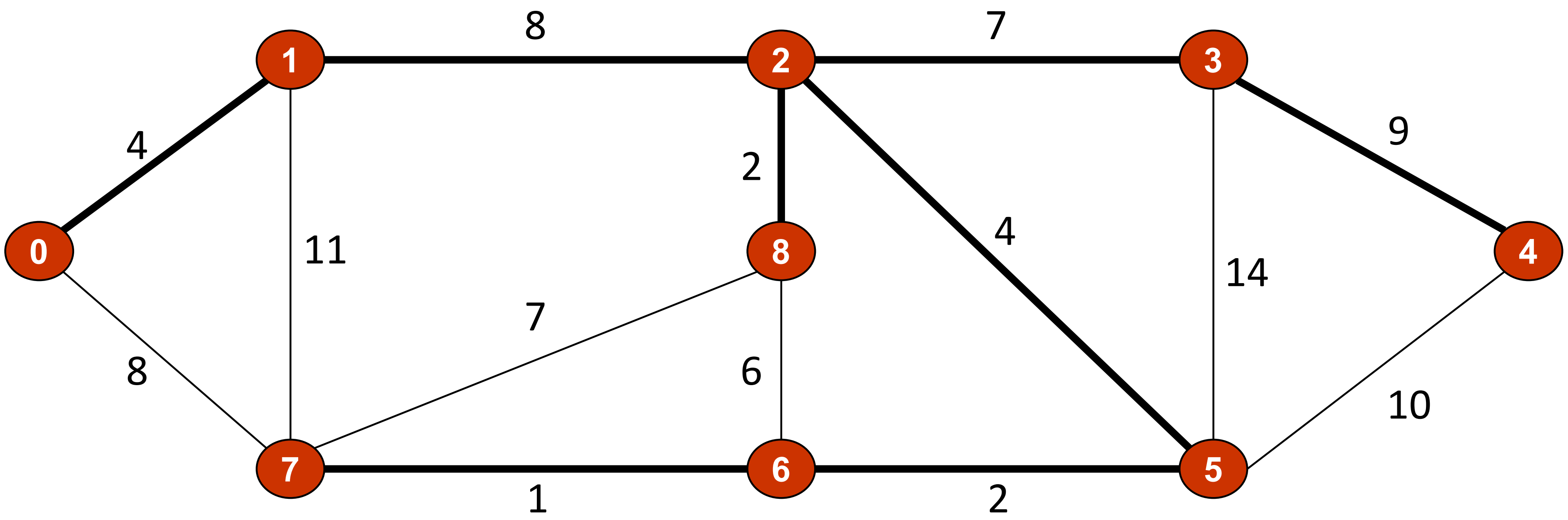
Prims Algorithm

- Επιλογή της σύνδεσης 2-3
  - $N_C \leftarrow \{0,1,2,8,5,6,7\} \cup \{3\}$
  - $E_S \leftarrow \{(0,1),(1,2),(2,8),(2,5),(5,6),(6,7)\} \cup \{(2,3)\}$



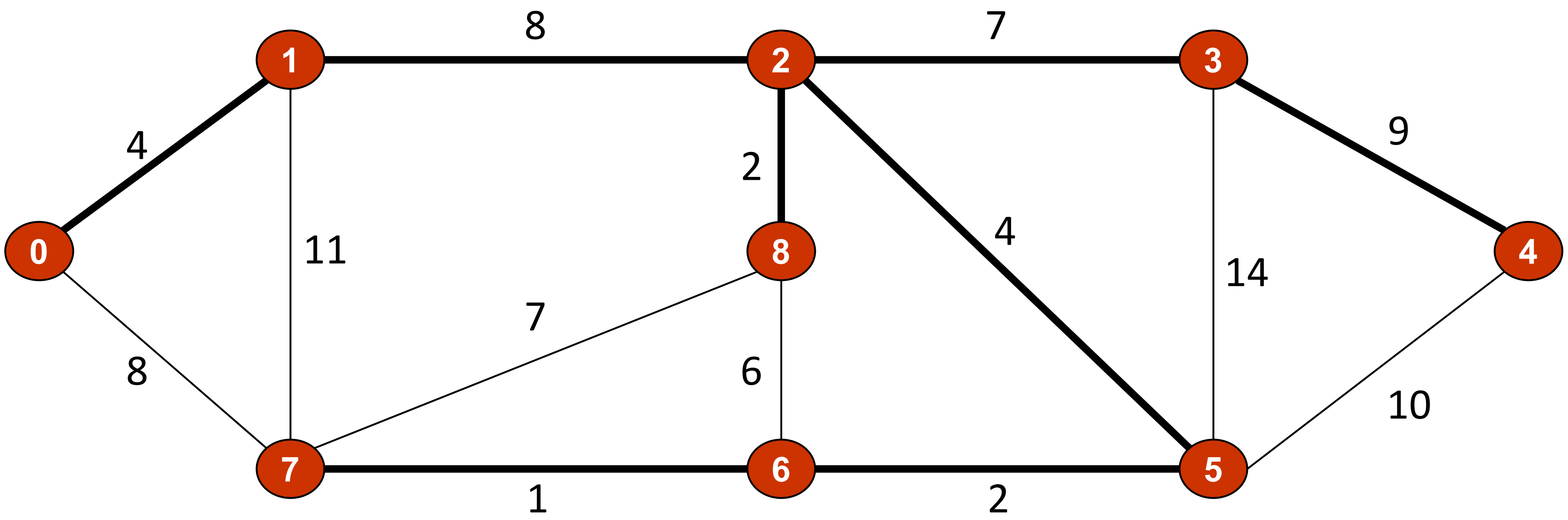
Prims Algorithm

- Επιλογή οικονομικότερης σύνδεσης



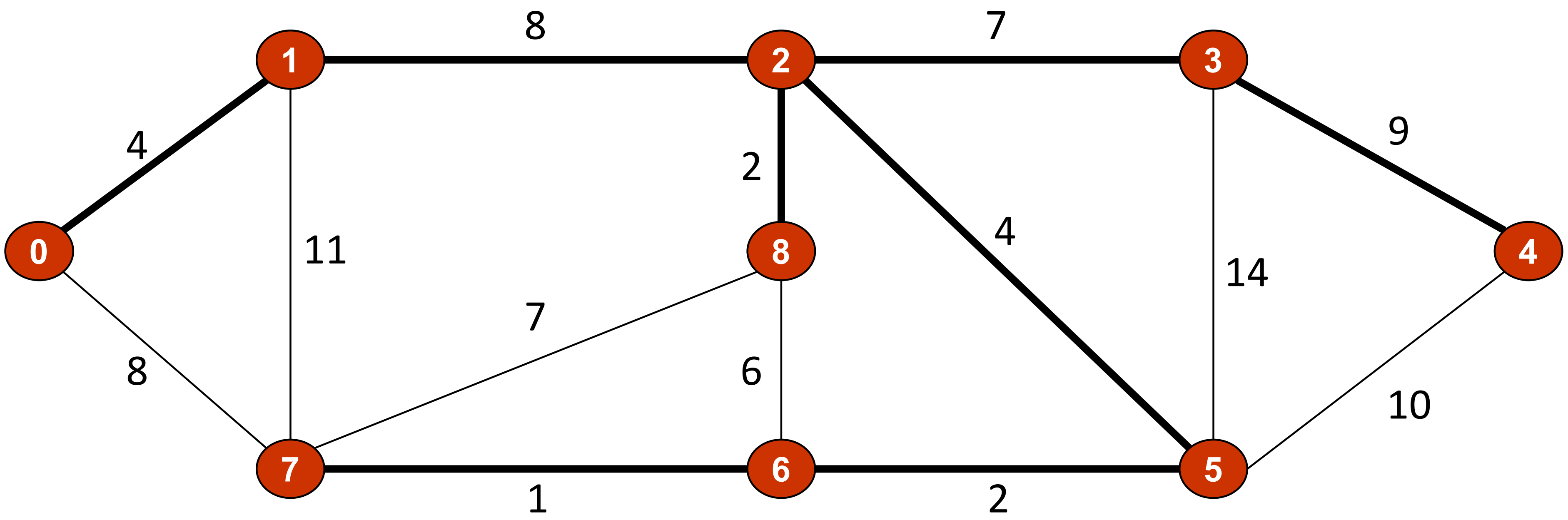
Prims Algorithm

- Επιλογή της σύνδεσης 3-4
  - $N_C \leftarrow \{0,1,2,8,5,6,7,3\} \cup \{3\}$
  - $E_S \leftarrow \{(0,1),(1,2),(2,8),(2,5),(5,6),(6,7),(2,3)\} \cup \{(3,4)\}$



Prims Algorithm

- $E_S \leftarrow \{(0,1), (1,2), (2,8), (2,5), (5,6), (6,7), (2,3), (3,4)\}$
- Τερματισμός Αλγορίθμου: $|E_S| = n - 1 = 9 - 1 = 8$
- Συνολικό Κόστος: 35



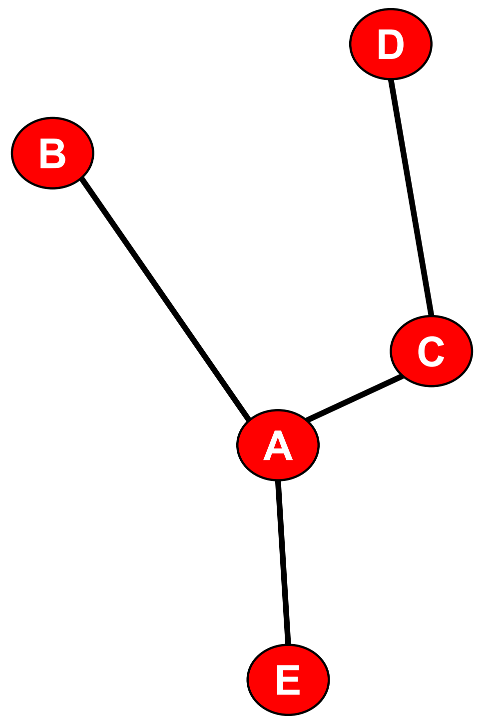
Kruskal's Algorithm

- *Τα βήματα εκτέλεσης του αλγόριθμου Kruskal είναι τα εξής*

1. *Διάταξη όλων των εν δυνάμει συνδέσεων*
2. *Για κάθε σύνδεση e*
   1. *Έλεγξε αν η επιλογή της σύνδεσης δημιουργεί κύκλο στο σχηματιζόμενο δέντρο*
   2. *Εφόσον δε δημιουργεί κύκλο, πρόσθεσε τη σύνδεση στο* E_S $(E_S \leftarrow E_S \cup e)$
   3. Αν $|E_S| = n - 1$, τερματισμός του αλγορίθμου

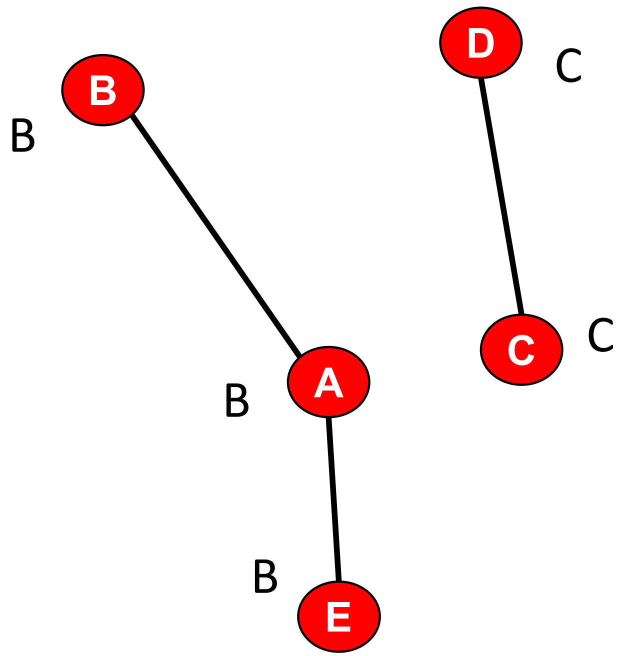
Kruskal's Algorithm

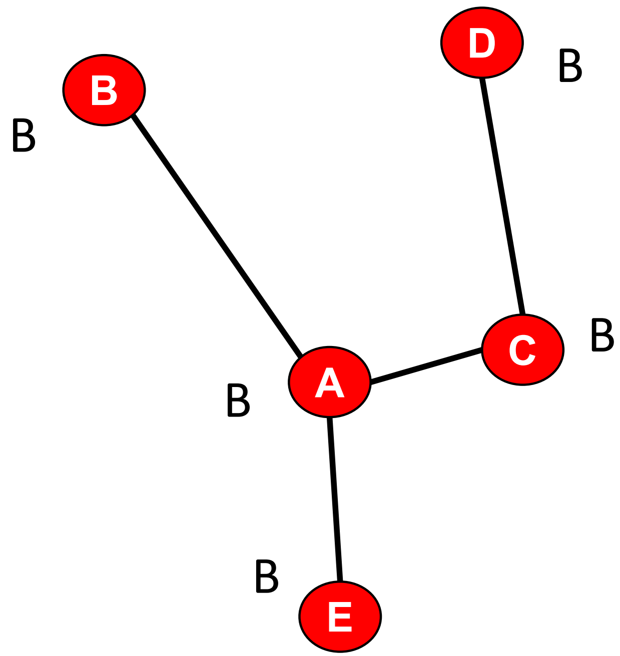
- Ο έλεγχος για το σχηματισμό κύκλου κατά την προσθήκη μίας ακμής είναι ένα απαιτητικό βήμα (εφόσον επιθυμούμε να γίνει με αποδοτικό τρόπο)
- Ας θεωρήσουμε ένα δέντρο
- Η προσθήκη της ακμής (B, D) θα δημιουργούσε έναν κύκλο
- Παρατηρούμε πως κάθε σύνδεση δύο κόμβων που ανήκουν στο ίδιο δέντρο δημιουργεί κύκλους
- Επομένως αν φτιάχναμε ένα αλγόριθμο ο οποίος ελέγχει τη δημιουργία κύκλων κατά την προσθήκη μίας σύνδεσης, στην ουσία θα πρέπει να ελέγχει αν οι δύο κόμβοι οι οποίοι ανήκουν στη σύνδεση ανήκουν στο ίδιο δέντρο
- Επομένως, θα πρέπει κάθε κόμβος να περιέχει μία πληροφορία η οποία να ορίζει μονοσήμαντα την ταυτότητα του δέντρου στο οποίο ανήκει
- Για να πραγματοποιήσουμε το στόχο αυτό, κάθε κόμβος ο οποίος ανήκει σε ένα δέντρο χαρακτηρίζεται από έναν κοινό κόμβο ο οποίος είναι ο εκπρόσωπος του δέντρου



Kruskal's Algorithm

- Η ανάγκη εκπροσώπησης κάθε κόμβου ο οποίος ανήκει σε ένα δέντρο δημιουργεί όμως το εξής πρόβλημα:
- Πως ανανεώνεται ο εκπρόσωπος των κόμβων οι οποίοι ανήκουν σε δύο δέντρα, όταν τα δέντρα αυτά ενώνονται;
- Π.χ. Όταν τα δύο δέντρα του σχήματος ενωθούν για να δημιουργήσουν ένα νέο δέντρο, θέλουμε όλοι οι κόμβοι του νέου (μεγαλύτερου) δέντρου να έχουν ένα κοινό εκπρόσωπο



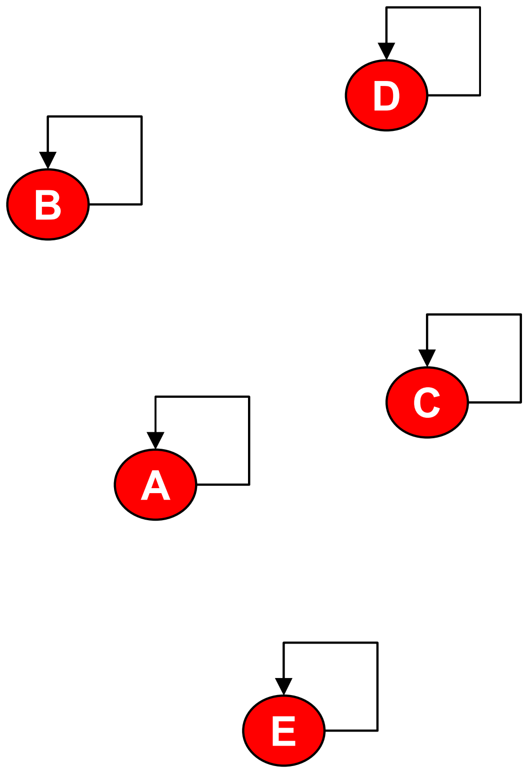


Kruskal's Algorithm

- Για να το κάνουμε αυτό χρησιμοποιείται η εξής λογική:
- Κάθε κόμβος έχει έναν «πατέρα» κόμβο
  - Όταν ο κόμβος δεν είναι ενωμένος με κανέναν άλλον, ο πατέρα είναι ο ίδιος του ο εαυτός
  - Ένας κόμβος είναι εκπρόσωπος ενός συνόλου κόμβων, όταν πατέρα του είναι ο εαυτός του
- Κάθε φορά που δύο κόμβοι A και B ενώνονται, βρίσκουμε τους εκπροσώπους τους $r(A)$ και $r(B)$ και τίθεται ως πατέρας ενός εκ των δύο εκπροσώπων, έστω του $r(A)$, ο άλλος εκπρόσωπος $r(B)$
- Έτσι, όποτε θέλουμε να βρούμε τον εκπρόσωπο ενός κόμβου αρχίζουμε να ακολουθούμε την αλληλουχία των «πατεράδων»
- Εντοπίζουμε τον εκπρόσωπο, όταν φτάσουμε στον κόμβο του οποίου ο πατέρας είναι ο εαυτός του

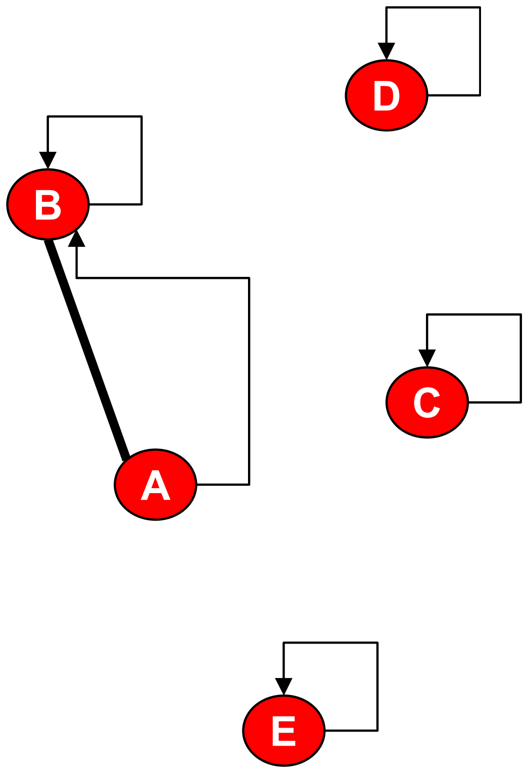
Kruskal’s Algorithm

- Αρχικά, οι κόμβοι δεν είναι συνδεδεμένοι, επομένως κάθε ένα από αυτούς, είναι ενωμένος με τον εαυτό του
- Επομένως, εκπρόσωπος του κάθε κόμβου είναι ο εαυτός του



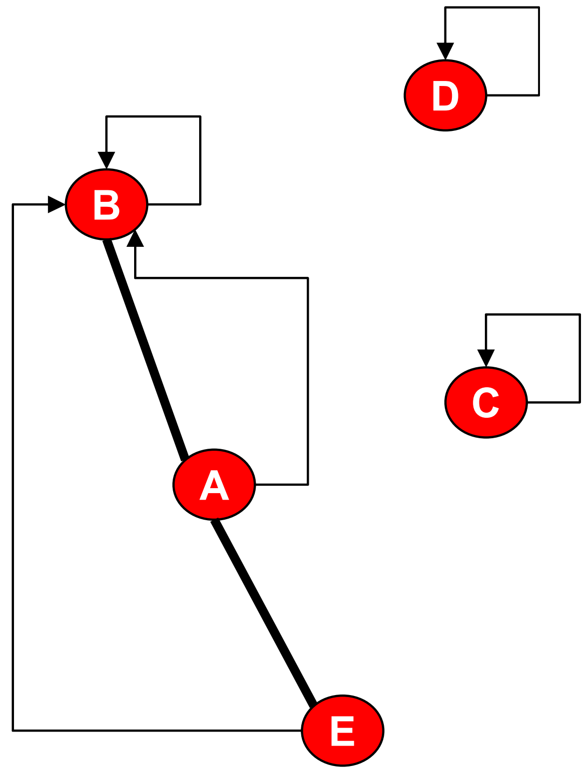
Kruskal's Algorithm

- Σύνδεση A με B
- Έστω πως συνδέεται ο κόμβος A και ο κόμβος B με την ακμή (A, B)
- Εκπρόσωπος του A: A
- Εκπρόσωπος του B: B
- Πατέρας του A: B



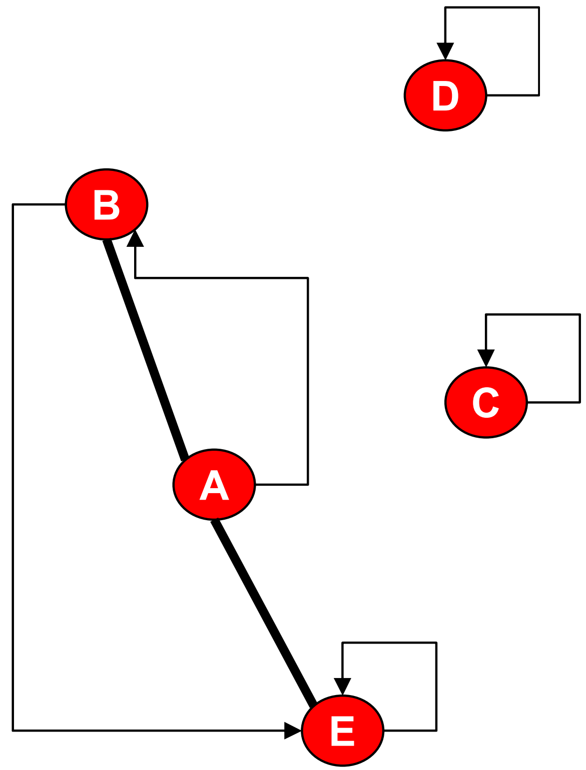
Kruskal's Algorithm

- Σύνδεση A με E
- Έστω πως συνδέεται ο κόμβος A και ο κόμβος E με την ακμή (A, E)
- Εκπρόσωπος του A: B
- Εκπρόσωπος του E: E
- Πατέρας του E: B
- Εκπρόσωπος των A B E: B



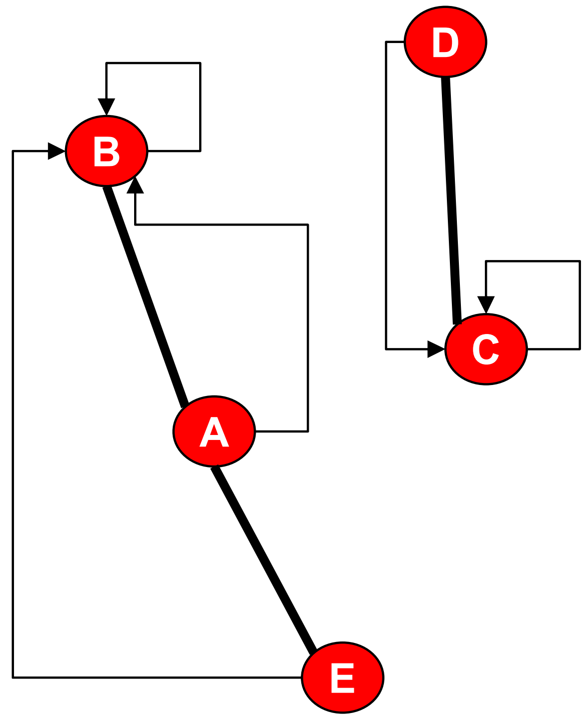
Kruskal's Algorithm

- **ΣΧΟΛΙΟ – Εναλλακτικός ορισμός πατέρα**
- Σύνδεση Α με Ε
- Έστω πως συνδέεται ο κόμβος Α και ο κόμβος Ε με την ακμή (Α, Ε)
- Εκπρόσωπος του Α: Β
- Εκπρόσωπος του Ε: Ε
- Πατέρας του Β: Ε
- Εκπρόσωπος των Α Β Ε: Ε
- Στην πράξη προτιμάμε την προηγούμενη ένωση καθώς, μειώνει το βάθος του δέντρου (Union by rank)



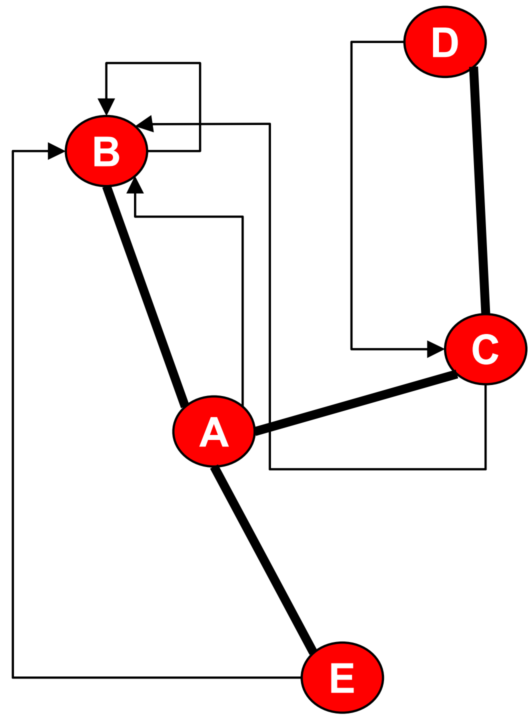
Kruskal's Algorithm

- Σύνδεση C με D
- Έστω πως συνδέεται ο κόμβος C και ο κόμβος D με την ακμή (C, D)
- Εκπρόσωπος του C: C
- Εκπρόσωπος του D: D
- Τίθεται πατέρας του D: C
- Εκπρόσωπος των A B C: B
- Εκπρόσωπος των D C: C

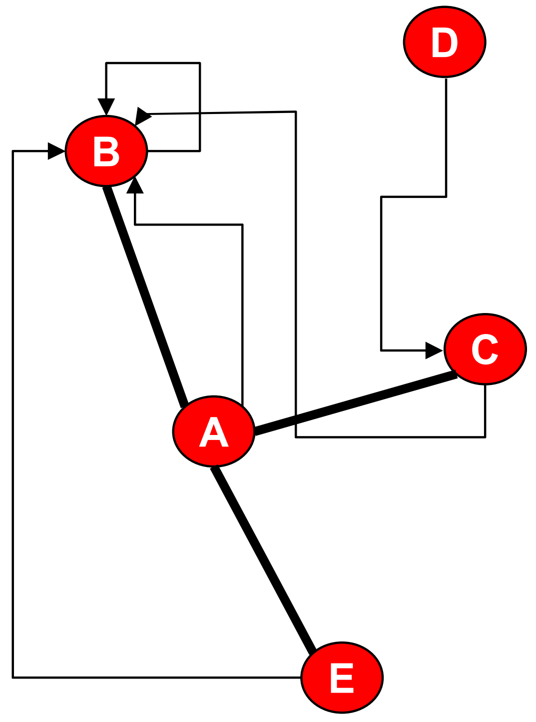


Kruskal’s Algorithm

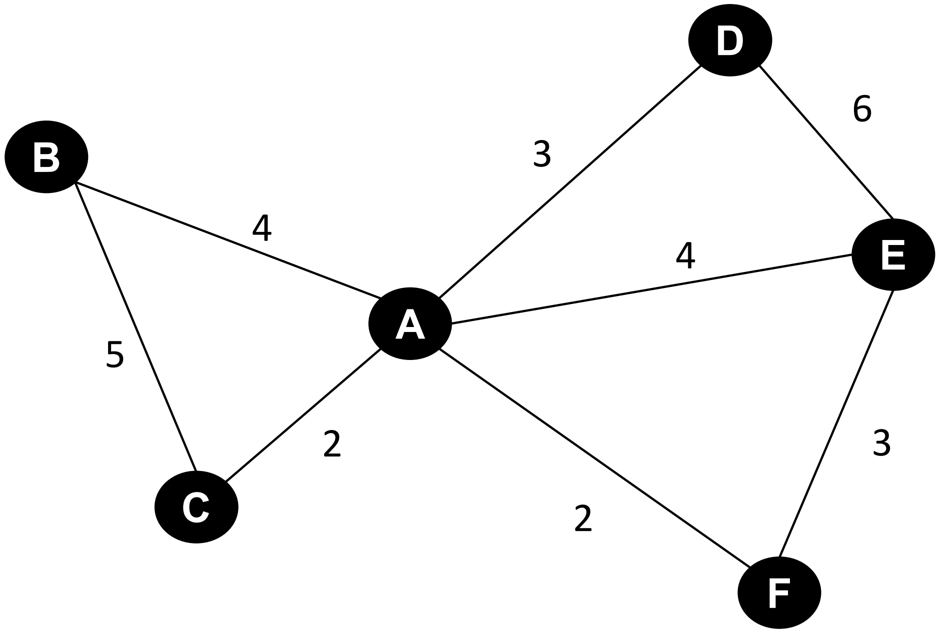
- Σύνδεση Α με C
- Έστω πως συνδέεται ο κόμβος Α και ο κόμβος C με την ακμή (A, C)
- Εκπρόσωπος του Α, ο Β
- Εκπρόσωπος του C, ο C
- Τίθεται πατέρας του C ο Β
- Εκπρόσωπος των A B C D E: ο κόμβος Β



Kruskal's Algorithm



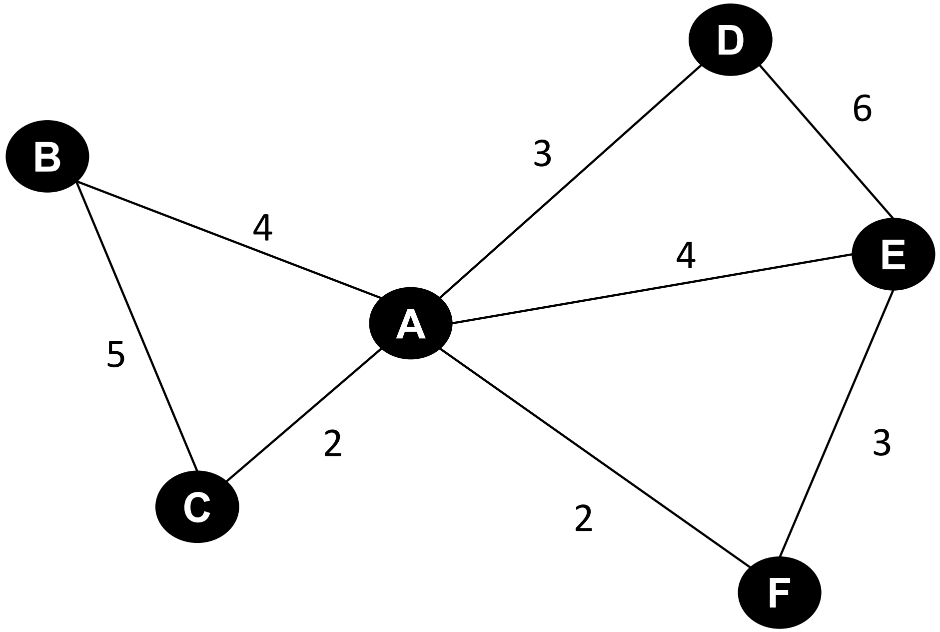
Kruskal's Algorithm



Kruskal's Algorithm

- Κατάταξη των Ακμών σε Αύξουσα σειρά κόστους

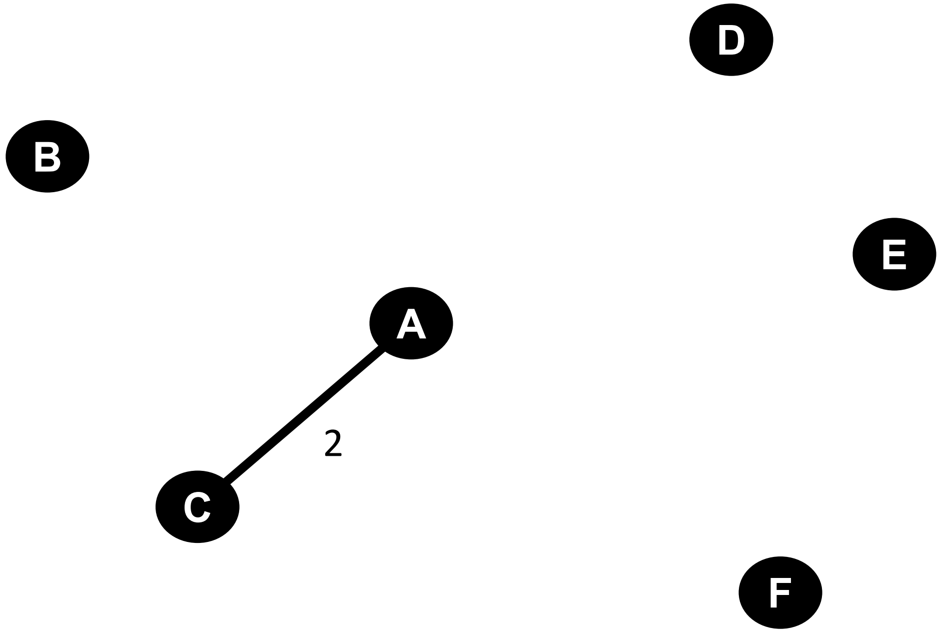
Ακμή	Κόστος
AC	2
AF	2
EF	3
AD	3
AE	4
AB	4
BC	5
DE	6



Kruskal's Algorithm

- Επιλογή AC

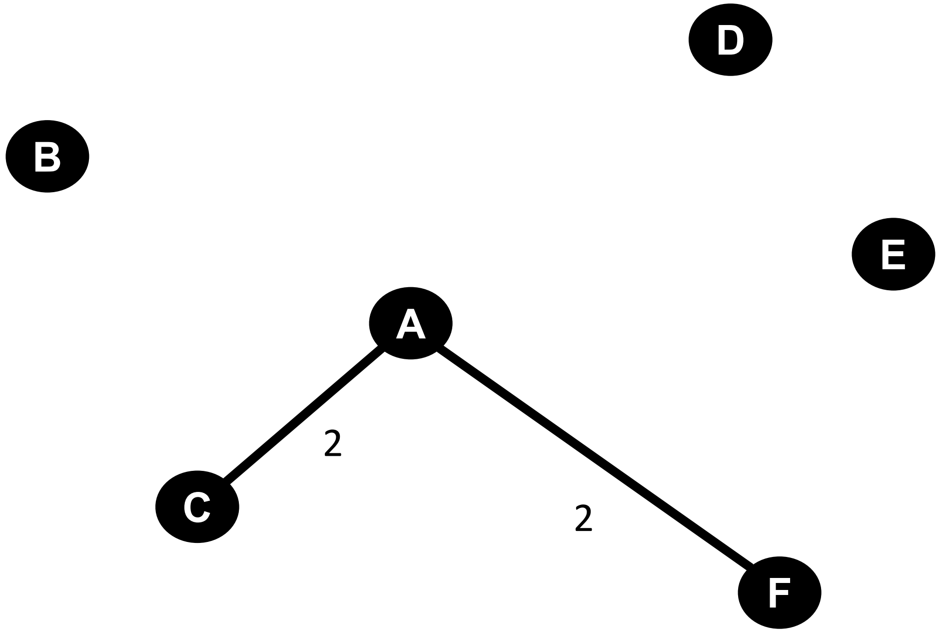
Ακμή	Κόστος
AC	**2**
AF	2
EF	3
AD	3
AE	4
AB	4
BC	5
DE	6



Kruskal's Algorithm

- Επιλογή AF

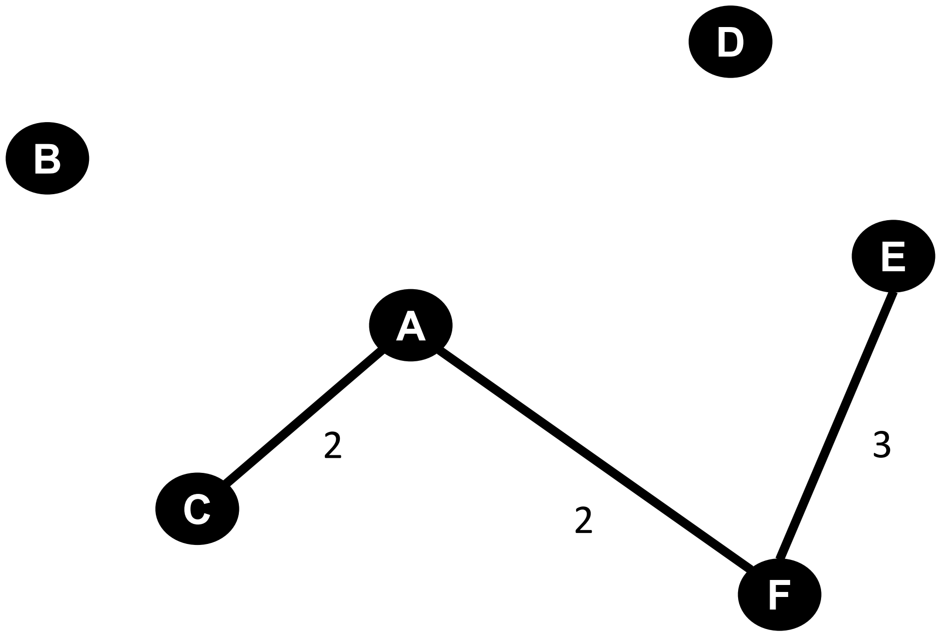
Ακμή	Κόστος
AC	**2**
AF	**2**
EF	3
AD	3
AE	4
AB	4
BC	5
DE	6



Kruskal's Algorithm

- Επιλογή EF

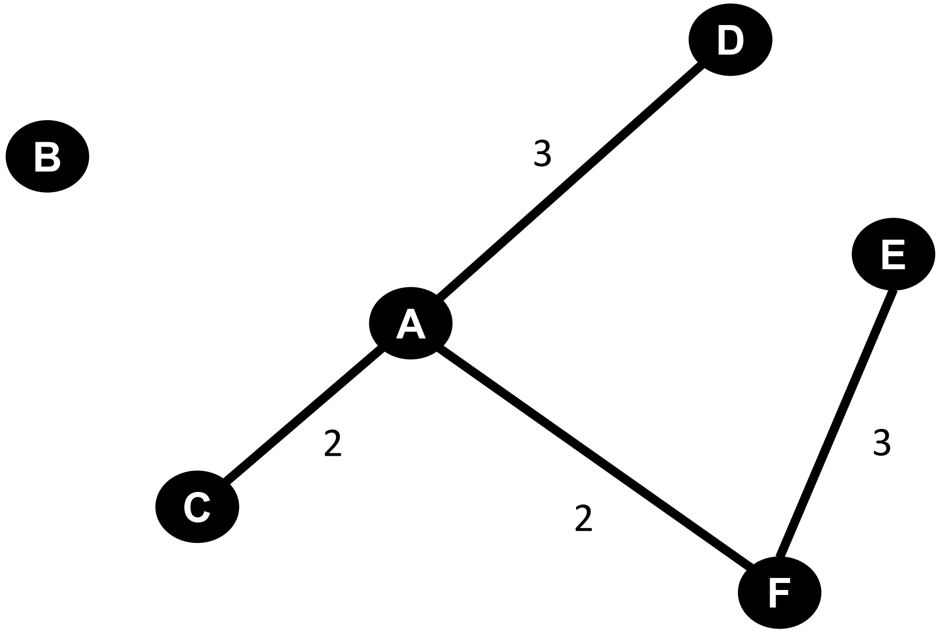
Ακμή	Κόστος
AC	**2**
AF	**2**
EF	**3**
AD	3
AE	4
AB	4
BC	5
DE	6



Kruskal's Algorithm

- Επιλογή AD

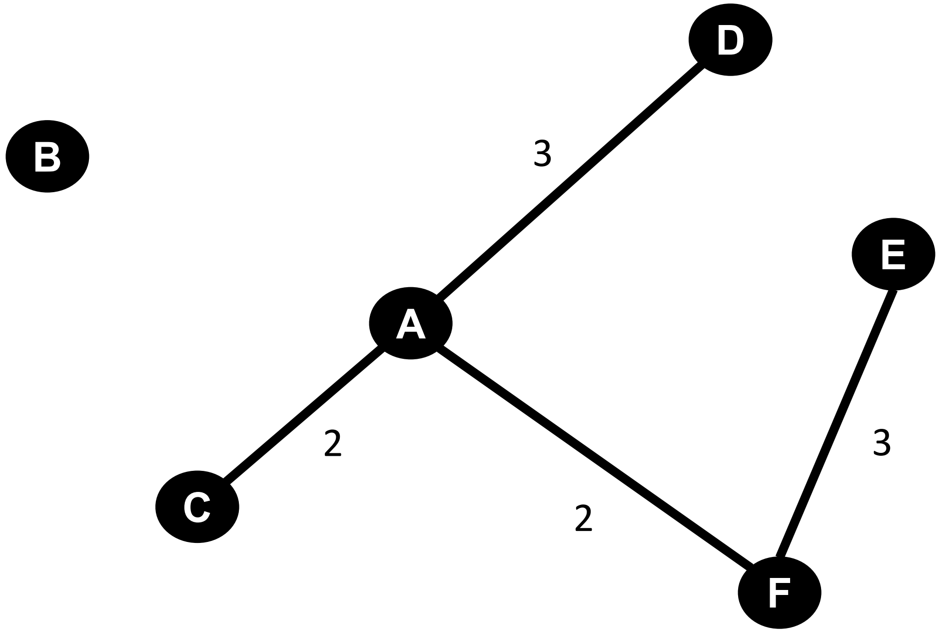
Ακμή	Κόστος
AC	**2**
AF	**2**
EF	**3**
AD	**3**
AE	4
AB	4
BC	5
DE	6



Kruskal's Algorithm

- Η Ακμή AE δεν επιλέγεται καθώς δημιουργεί κύκλο
- Ενώνει τους κόμβους A και E που ανήκουν στο ίδιο δέντρο

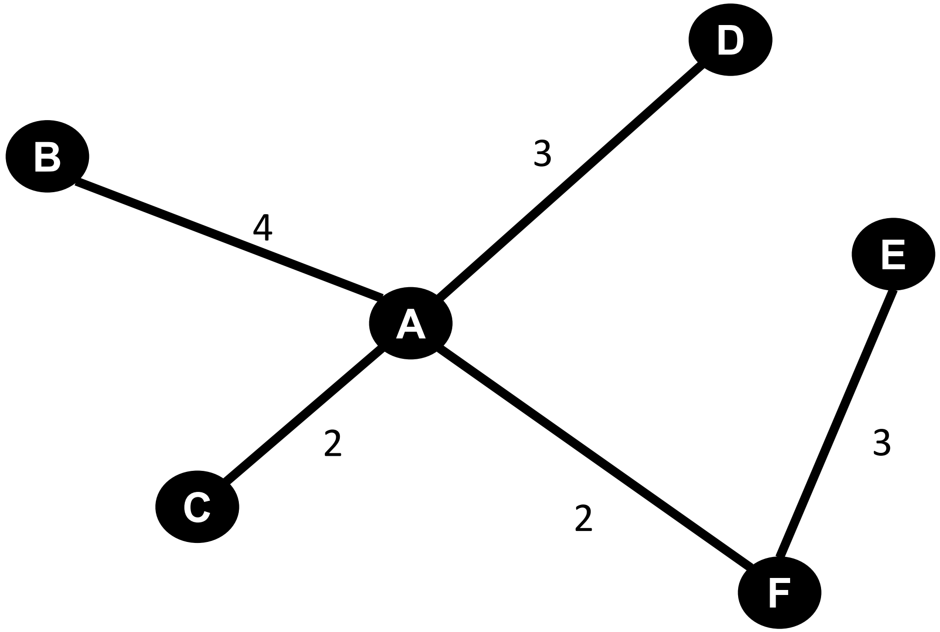
Ακμή	Κόστος
AC	**2**
AF	**2**
EF	**3**
AD	**3**
AE	4
AB	4
BC	5
DE	6



Kruskal's Algorithm

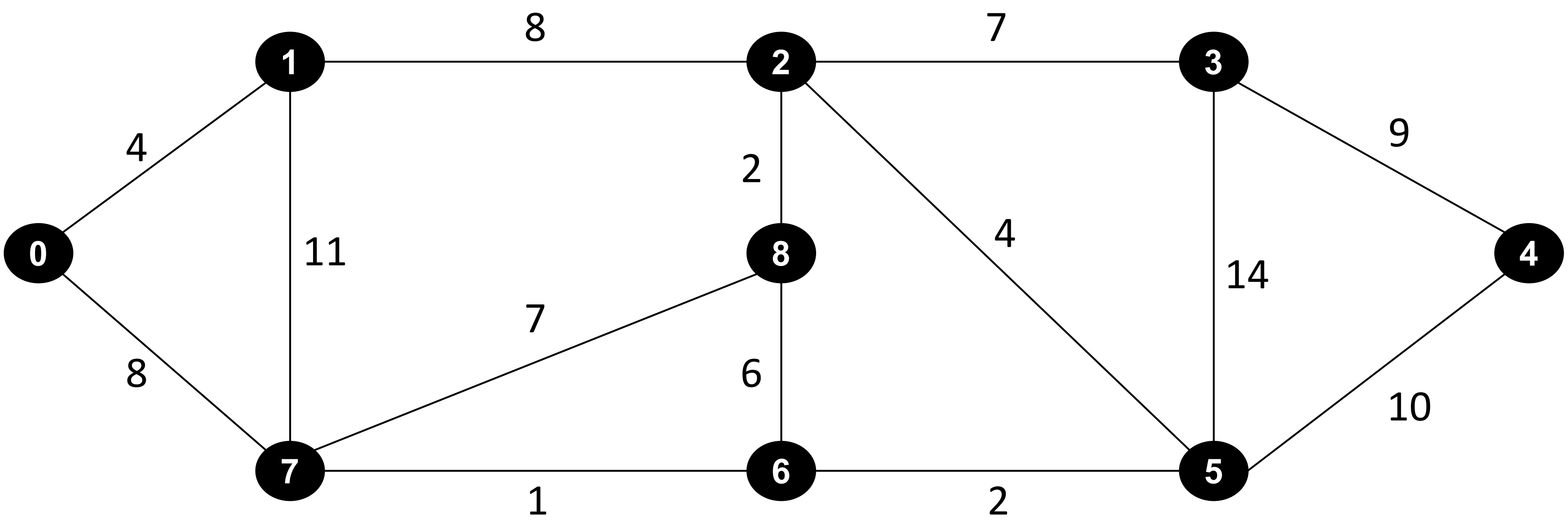
- Επιλογή AB
- Τερματισμός Αλγορίθμου
- Τελικό Κόστος: 14

Ακμή	Κόστος
AC	**2**
AF	**2**
EF	**3**
AD	**3**
AE	4
AB	**4**
BC	5
DE	6



Kruska’s Algorithm

- Εφαρμόστε τον αλγόριθμο Kruskal στο παρακάτω γράφο, ώστε να σχηματιστεί το Minimum Spanning Tree του συνόλου των 9 κόμβων



Kruska's Algorithm

- Εφαρμόστε τον αλγόριθμο Kruskal στο παρακάτω γράφο, ώστε να σχηματιστεί το Minimum Spanning Tree του συνόλου των 9 κόμβων

(6,7):1	(5,6):2	(2,8):2	(0,1):4	(2,5):4	(6,8):6	(2,3):7
(7,8):7	(1,2):8	(0,7):8	(3,4):9	(4,5):10	(1,7):11	(3,5):14