ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ

Θεωρία και Αλγόριθμοι Γράφων

Ενότητα # **7**: Ελάχιστα Ζευγνύοντα Δένδρα

Ιωάννης Μανωλόπουλος
Τμήμα Πληροφορικής

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ

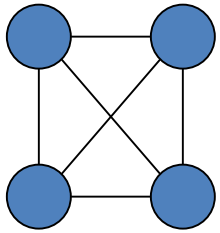
Ελάχιστα Ζευγνύοντα Δένδρα

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

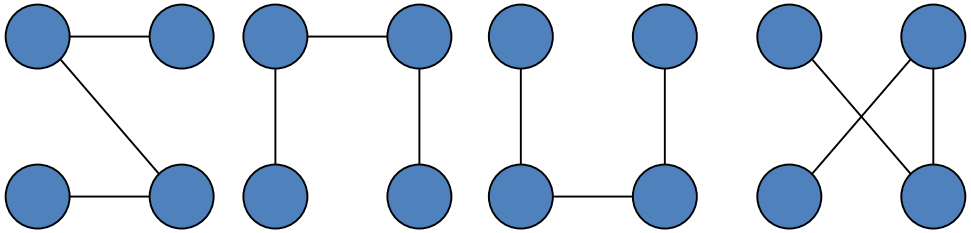
Ζευγνύοντα Δένδρα

- Ένα ζευγνύον δένδρο ενός γράφου είναι απλώς ένας υπογράφος που περιέχει όλες τις κορυφές του γράφου και είναι δένδρο.

Γράφος Α



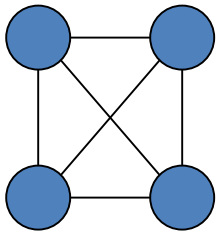
Μερικά Ζευγνύοντα Δένδρα ενός Γράφου Α



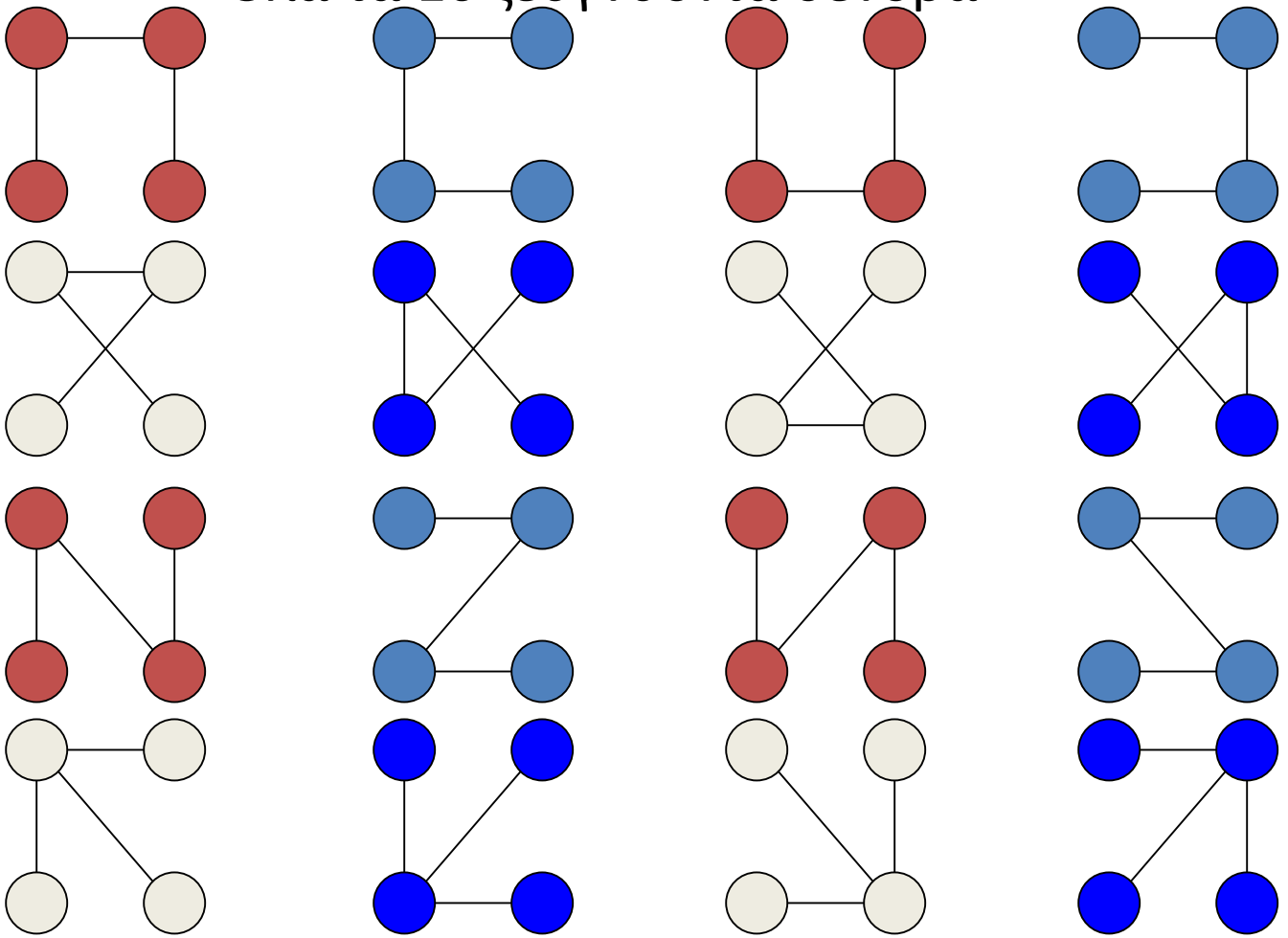
- Ένας γράφος μπορεί να έχει πολλά ζευγνύοντα δένδρα.

Ζευγνύοντα Δένδρα – Παράδειγμα

Πλήρης Γράφος



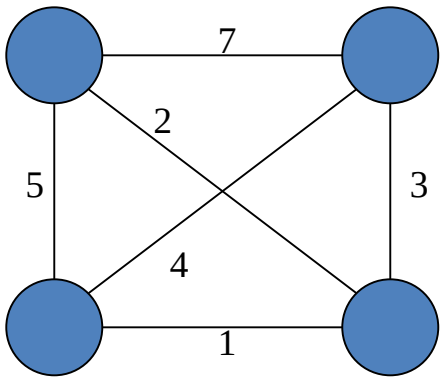
Όλα τα 16 ζευγνύοντα δένδρα



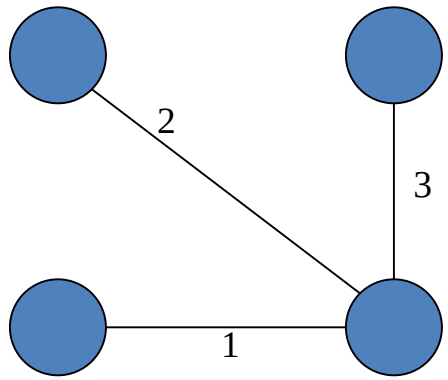
Ελάχιστα Ζευγνύοντα Δένδρα

- Το Ελάχιστο Ζευγνύον Δένδρο ενός γράφου είναι το Ζευγνύον Δένδρο ελάχιστου κόστους για το συγκεκριμένο γράφο.

Πλήρης Γράφος



Ελάχιστο Ζευγνύον Δένδρο



Αλγόριθμοι εύρεσης Ελάχιστου Ζευγνύοντος Δένδρου

- Αλγόριθμος Kruskal
- Αλγόριθμος Prim
- Αλγόριθμος Boruvka

Αλγόριθμος Kruskal I

- Ο αλγόριθμος δημιουργεί ένα δάσος από δένδρα.
- Αρχικά το δάσος αποτελείται από n δένδρα μίας κορυφής (χωρίς ακμές).
- Σε κάθε βήμα, προσθέτουμε μία ακμή (την πιο «φθηνή») που να ενώνει τα δύο δένδρα μαζί.
- Εάν πρόκειται να δημιουργήσει κύκλο η ακμή αυτή, θα ένωνε δύο κόμβρους που είναι ήδη μέρος ενός συνδεδεμένου δένδρου. Επομένως η ακμή αυτή δεν χρειάζεται.

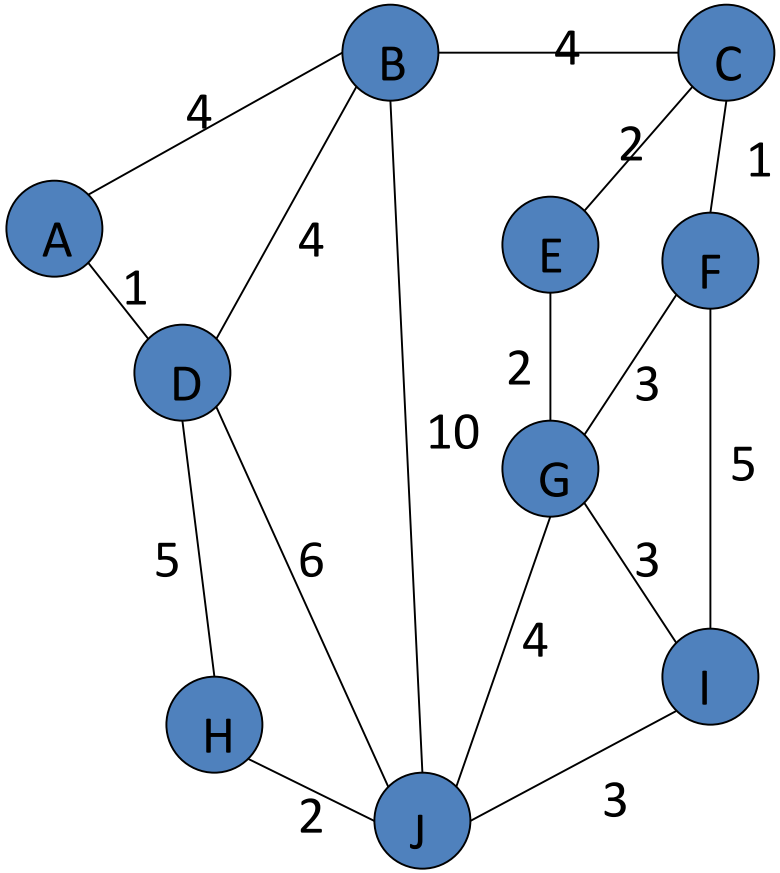
Αλγόριθμος Kruskal II

Τα βήματα είναι:

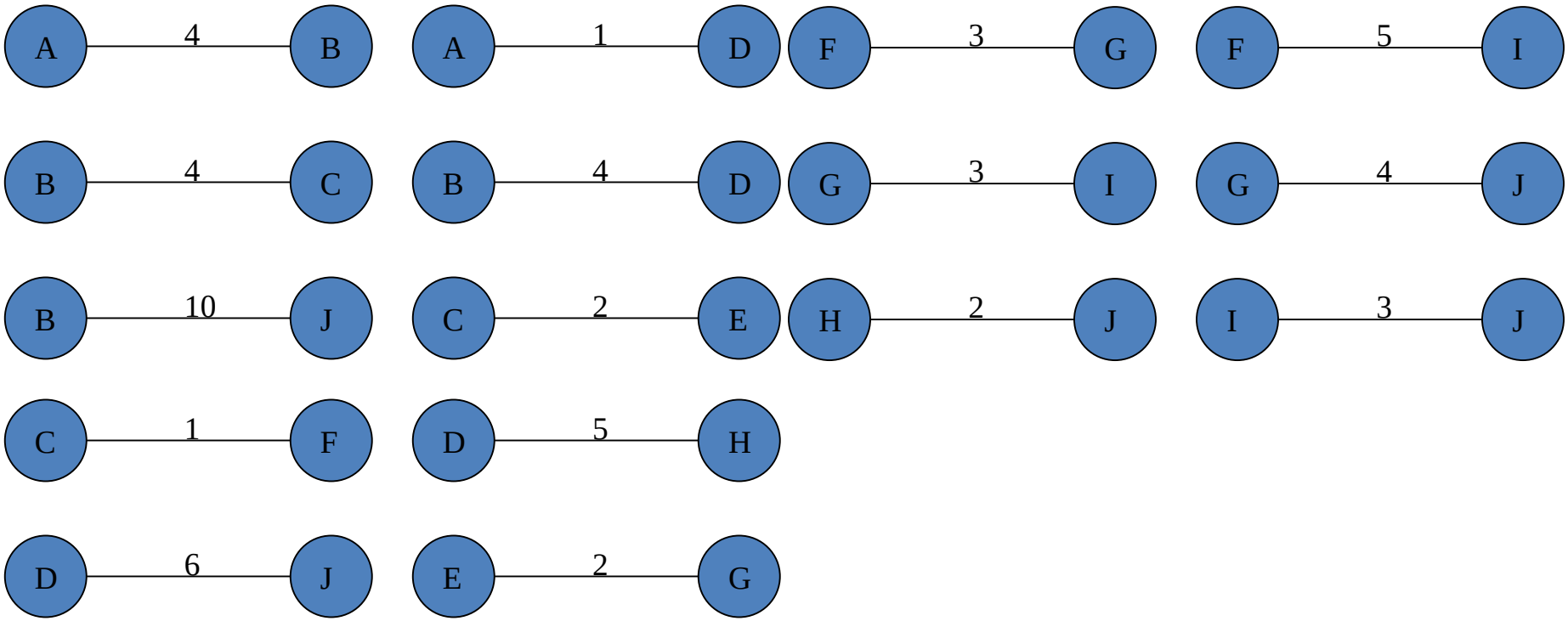
1. Κατασκευάζεται το δένδρο με κάθε κόμβο σε ξεχωριστό δένδρο.
2. Οι ακμές τοποθετούνται σε ουρά προτεραιότητας.
3. Μέχρι να προσθέσουμε n-1 ακμές,
    1. Βγάλε την πιο φτηνή ακμή από την ουρά,
    2. Εάν δημιουργεί κύκλο, την απορρίπτεις,
    3. Αλλιώς πρόσθεσε τη στο δάσος, ενώνοντας δύο δένδρα

Κάθε βήμα θα ενώνει δύο δένδρα στο δάσος, έτσι ώστε στο τέλος θα μείνει ένα δένρο στο Τ.

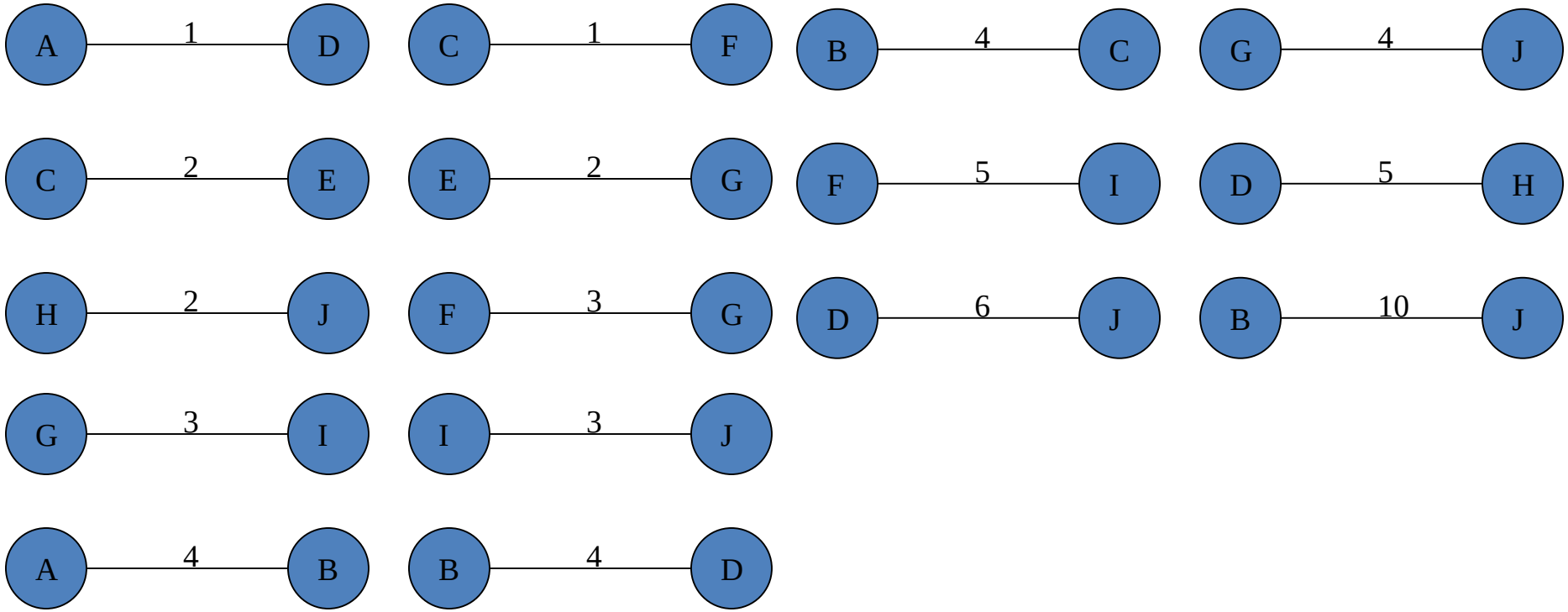
Αλγόριθμος Kruskal – Παράδειγμα Ι



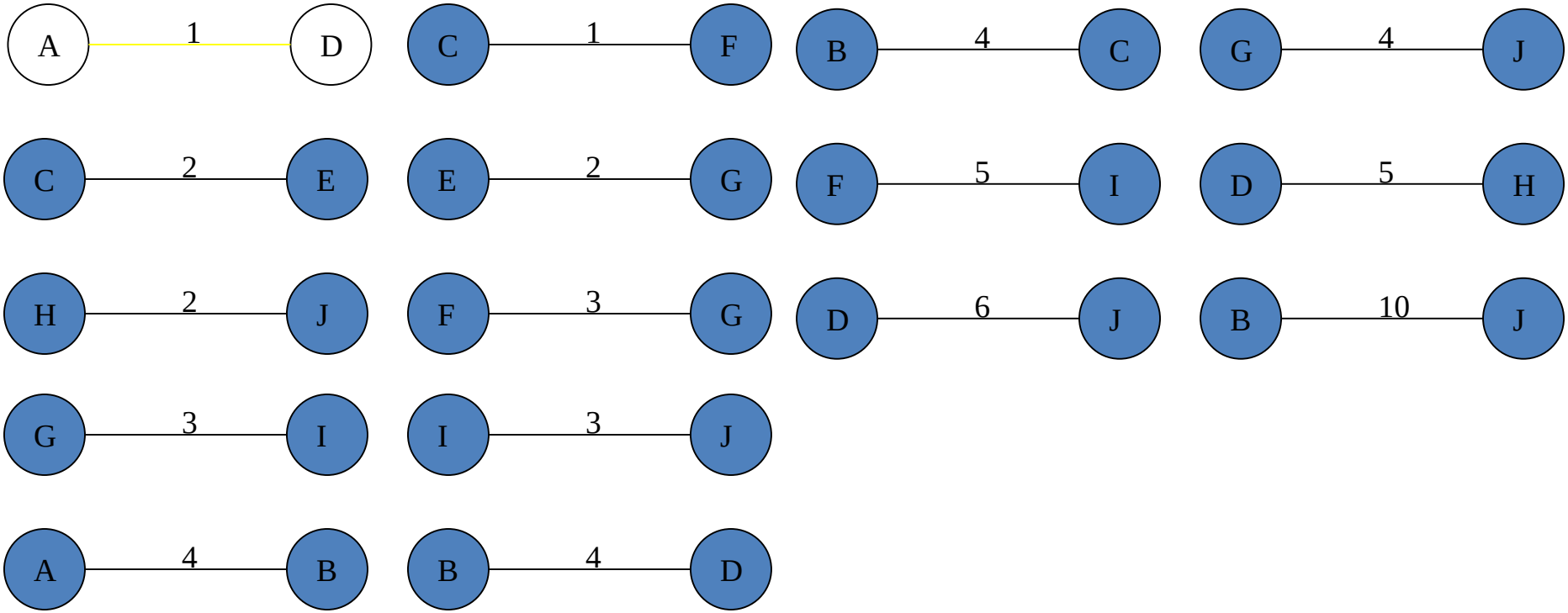
Αλγόριθμος Kruskal – Παράδειγμα II



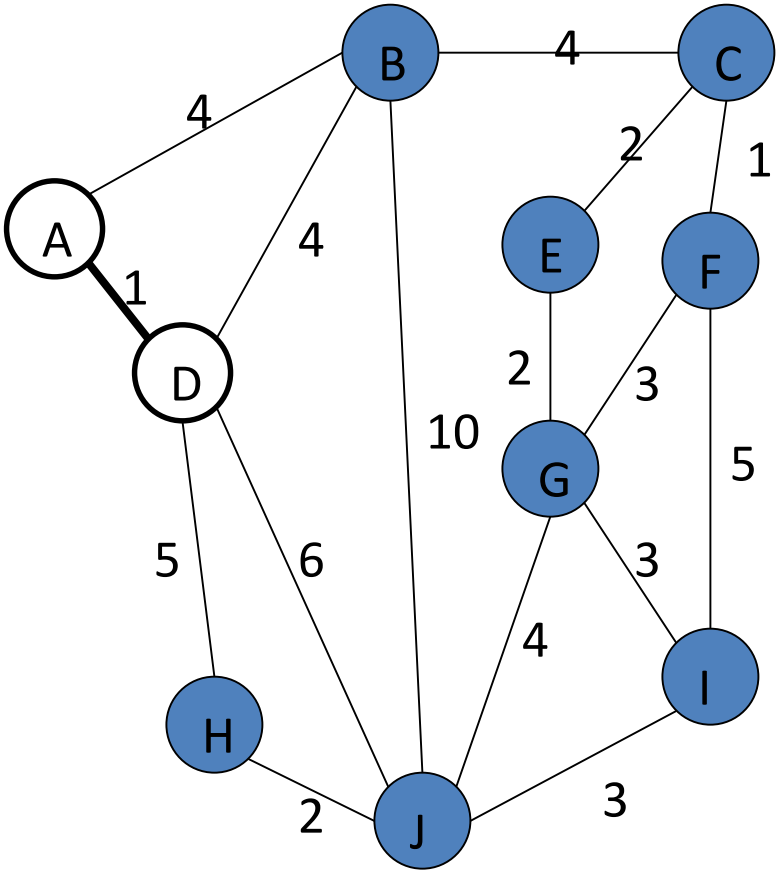
Αλγόριθμος Kruskal – Παράδειγμα III
Ταξινόμηση ακμών



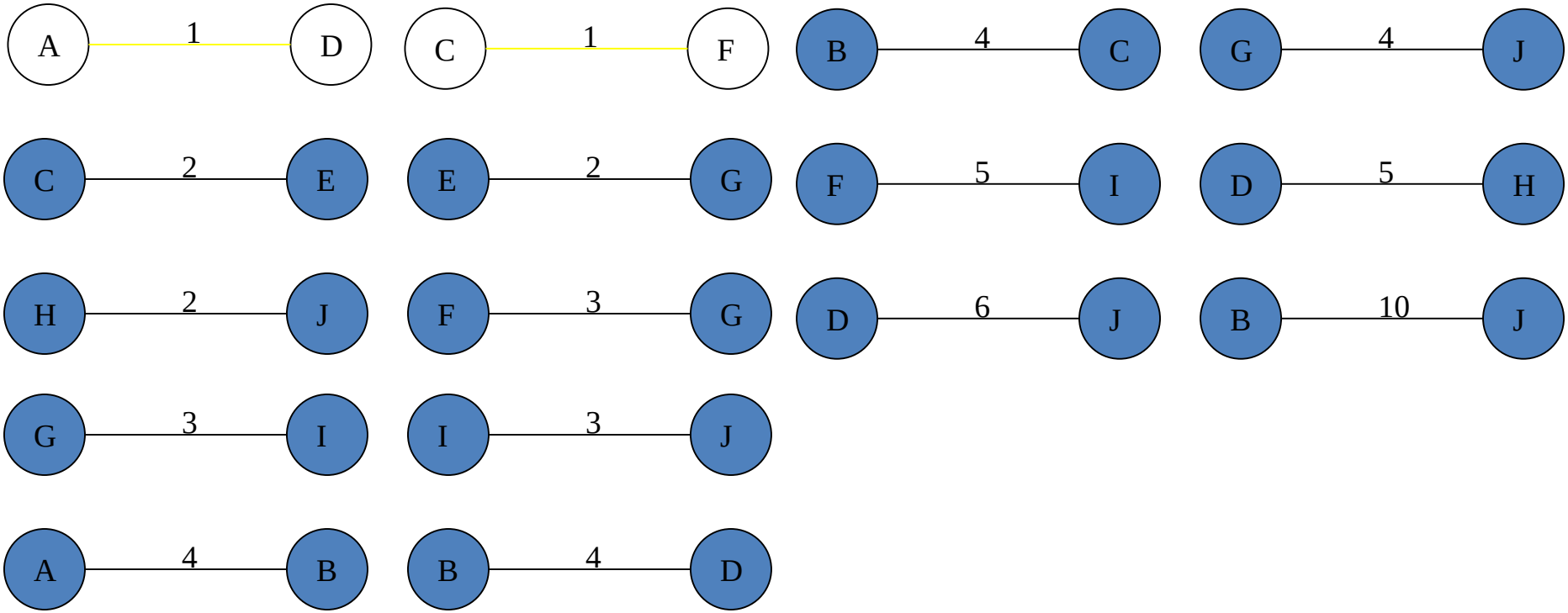
Αλγόριθμος Kruskal – Παράδειγμα IV
Προσθήκη ακμής



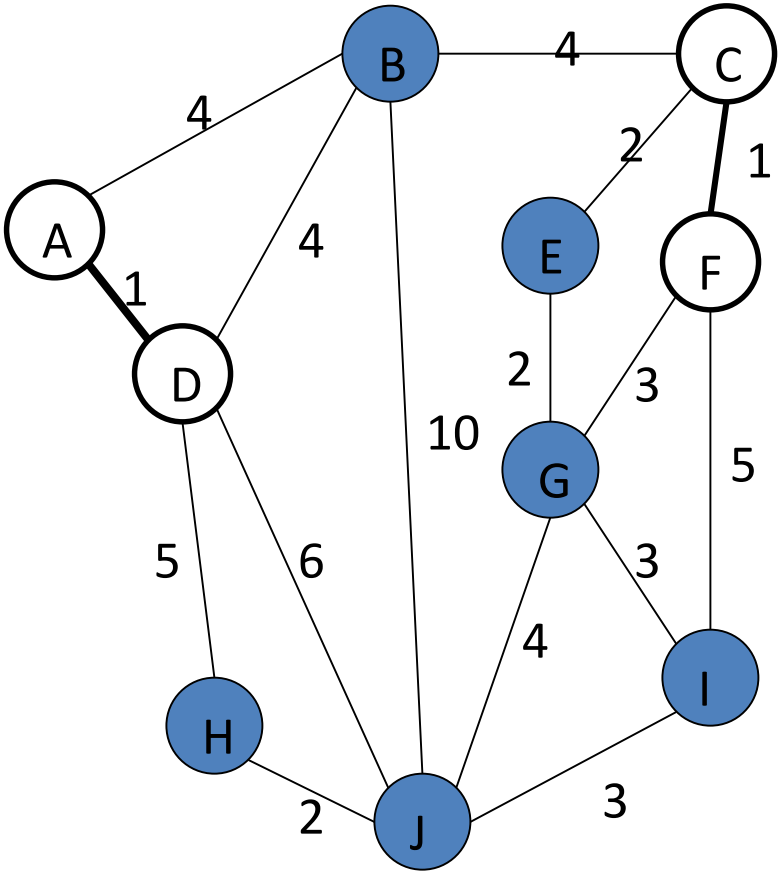
Αλγόριθμος Kruskal – Παράδειγμα V
Προσθήκη ακμής



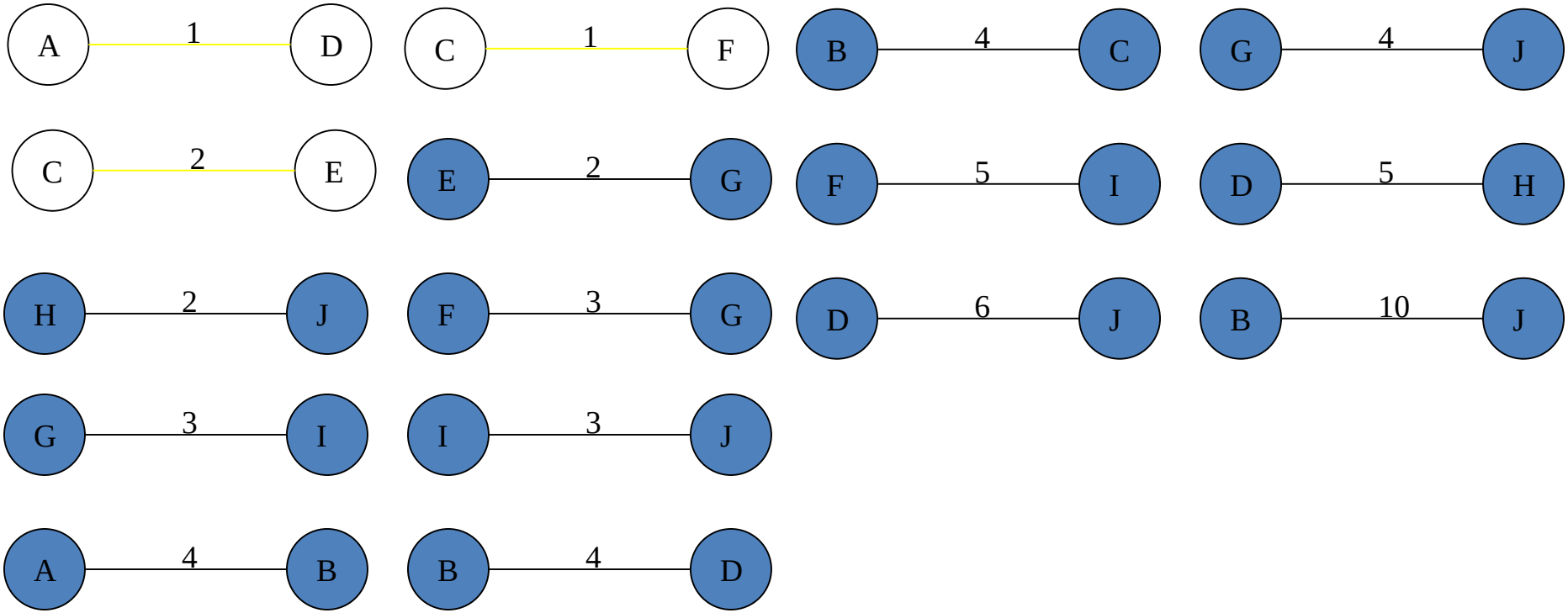
Αλγόριθμος Kruskal – Παράδειγμα VI
Προσθήκη ακμής



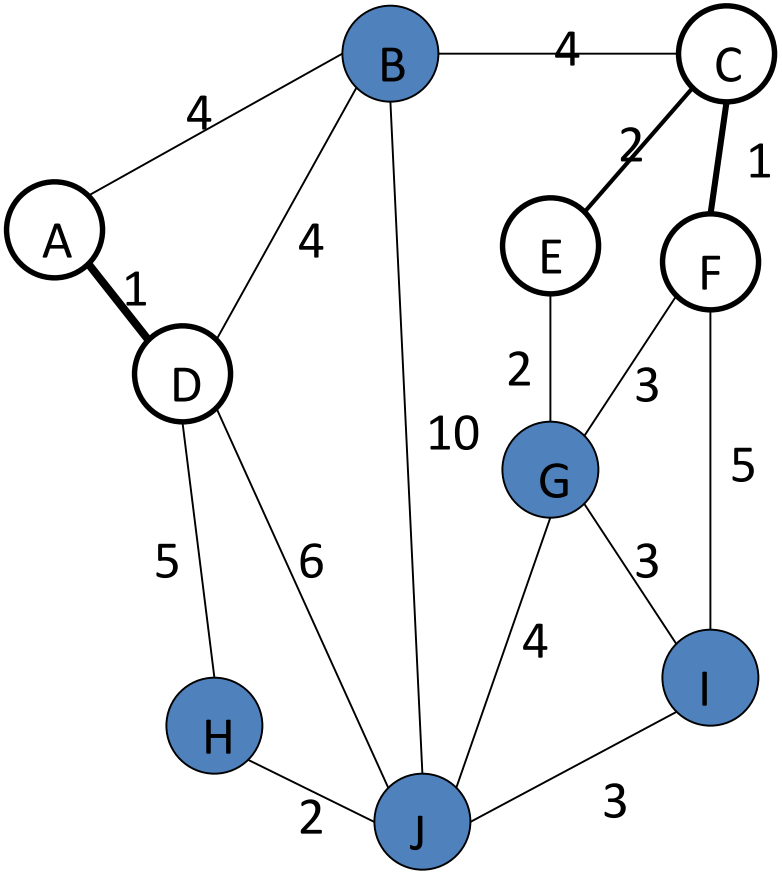
Αλγόριθμος Kruskal – Παράδειγμα VII
Προσθήκη ακμής



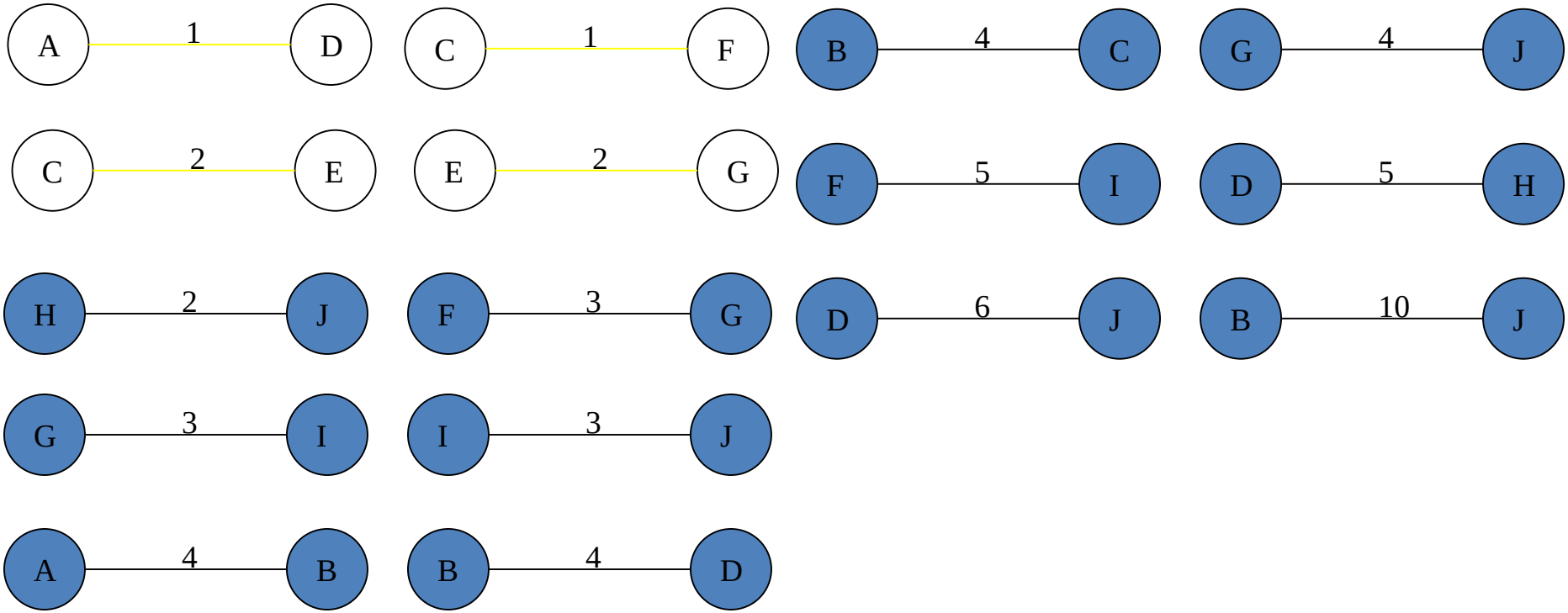
Αλγόριθμος Kruskal – Παράδειγμα VIII
Προσθήκη ακμής



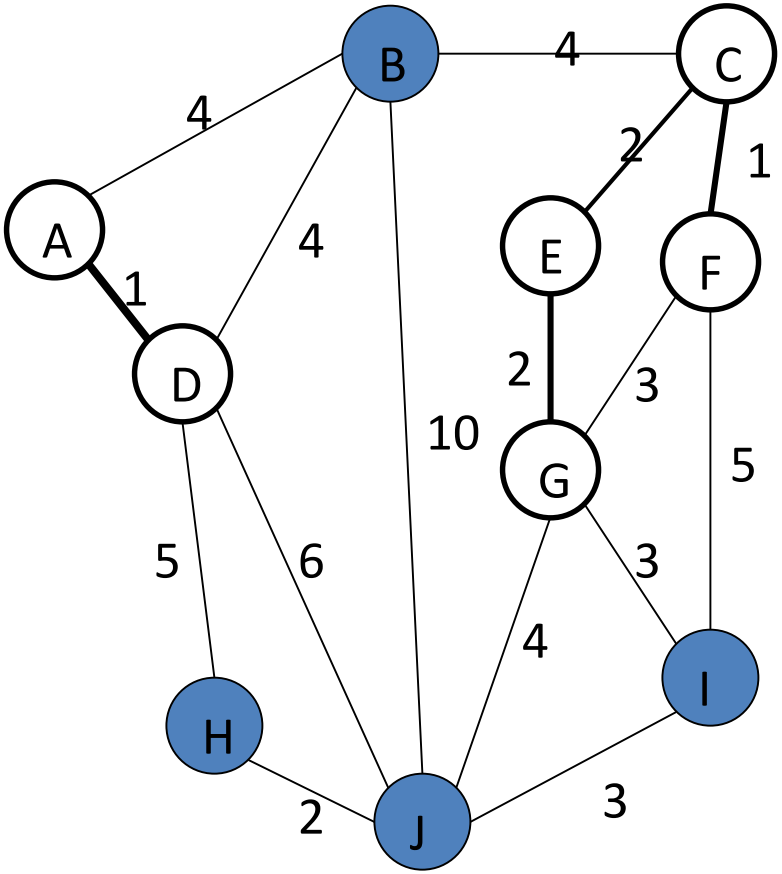
Αλγόριθμος Kruskal – Παράδειγμα IX
Προσθήκη ακμής



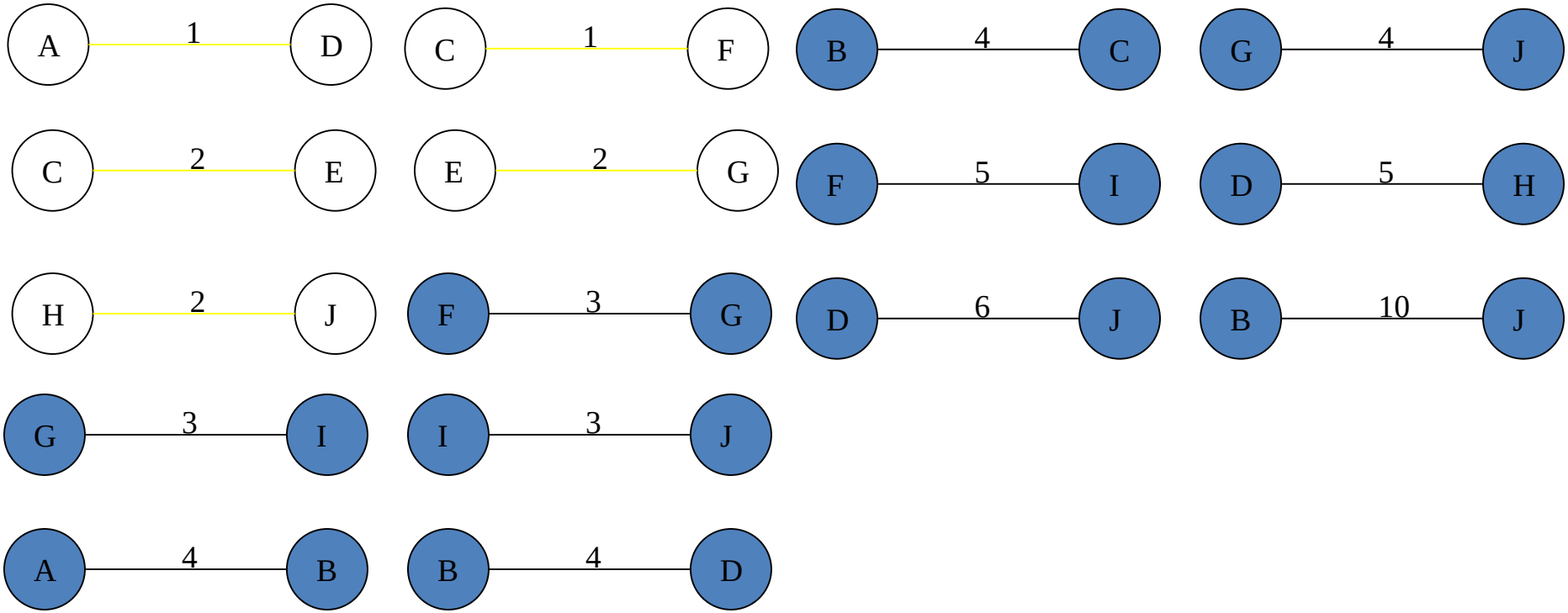
Αλγόριθμος Kruskal – Παράδειγμα Χ
Προσθήκη ακμής



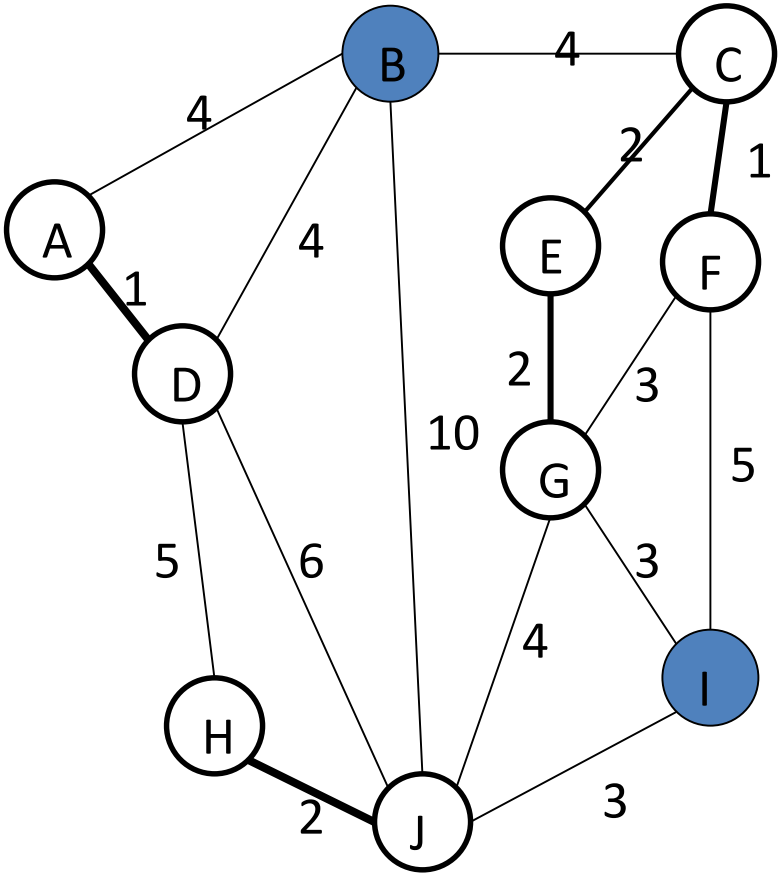
Αλγόριθμος Kruskal – Παράδειγμα XI
Προσθήκη ακμής



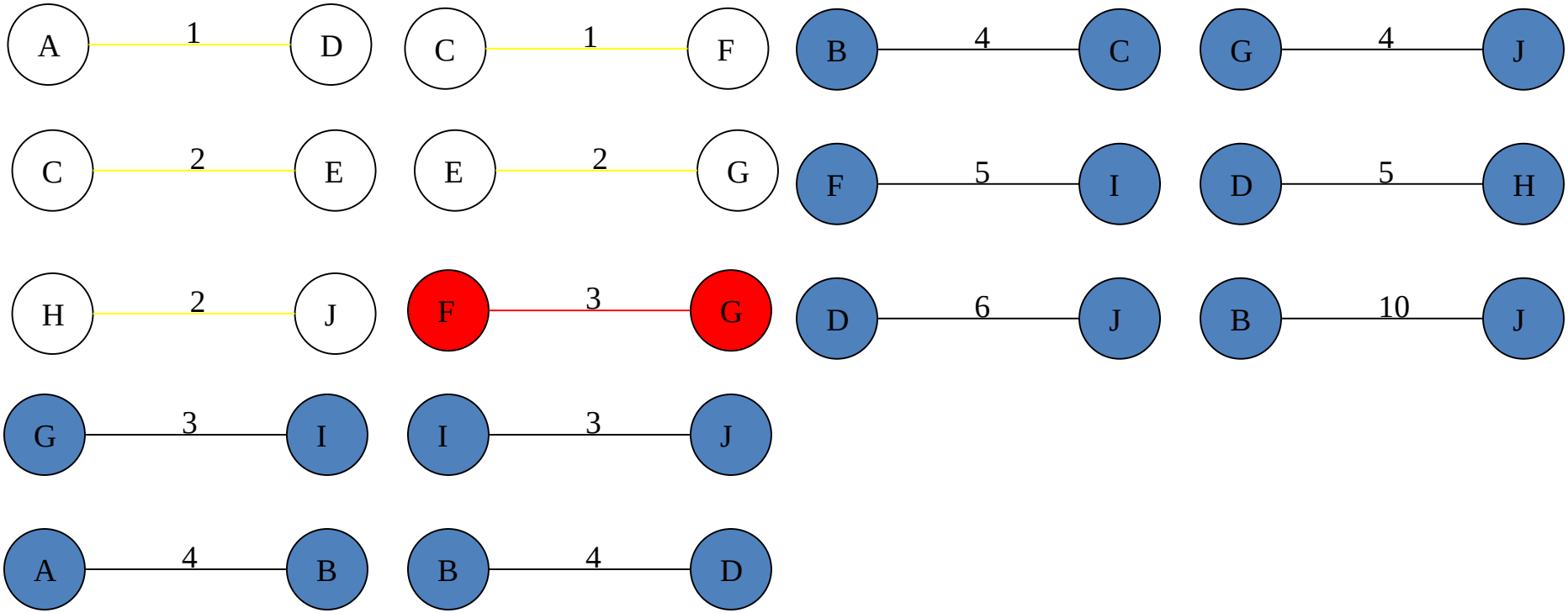
Αλγόριθμος Kruskal – Παράδειγμα XII
Προσθήκη ακμής



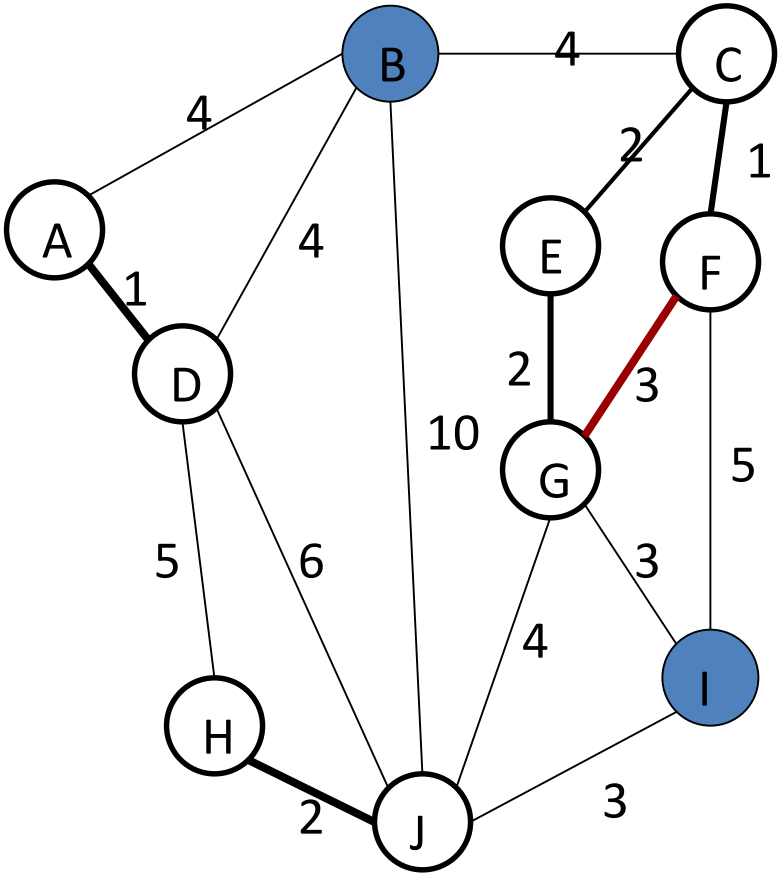
Αλγόριθμος Kruskal – Παράδειγμα XIII
Προσθήκη ακμής



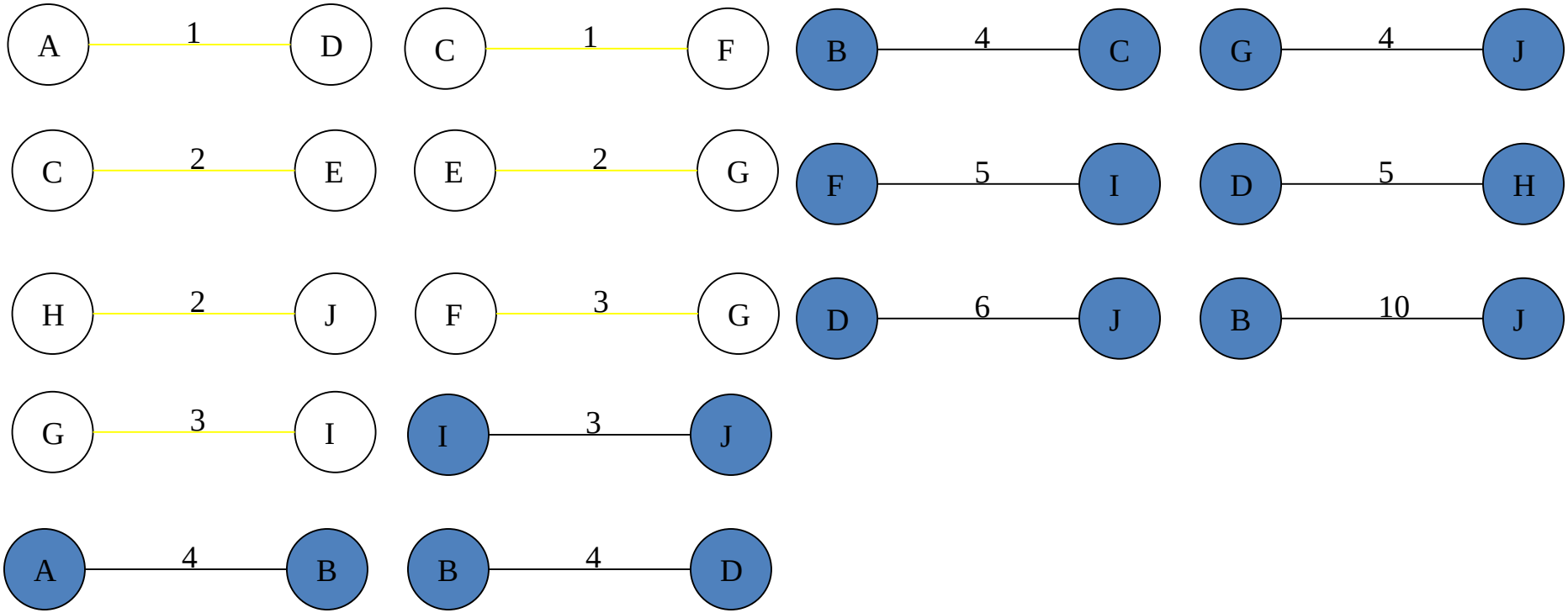
Αλγόριθμος Kruskal – Παράδειγμα XIV
Δημιουργία κύκλου



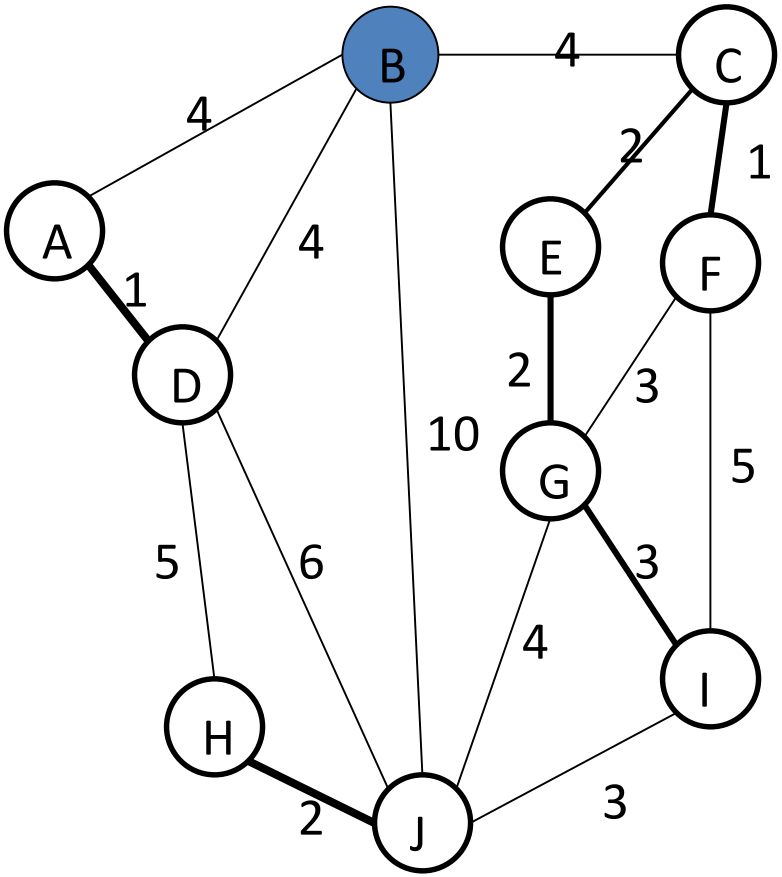
Αλγόριθμος Kruskal – Παράδειγμα XV
Δημιουργία κύκλου



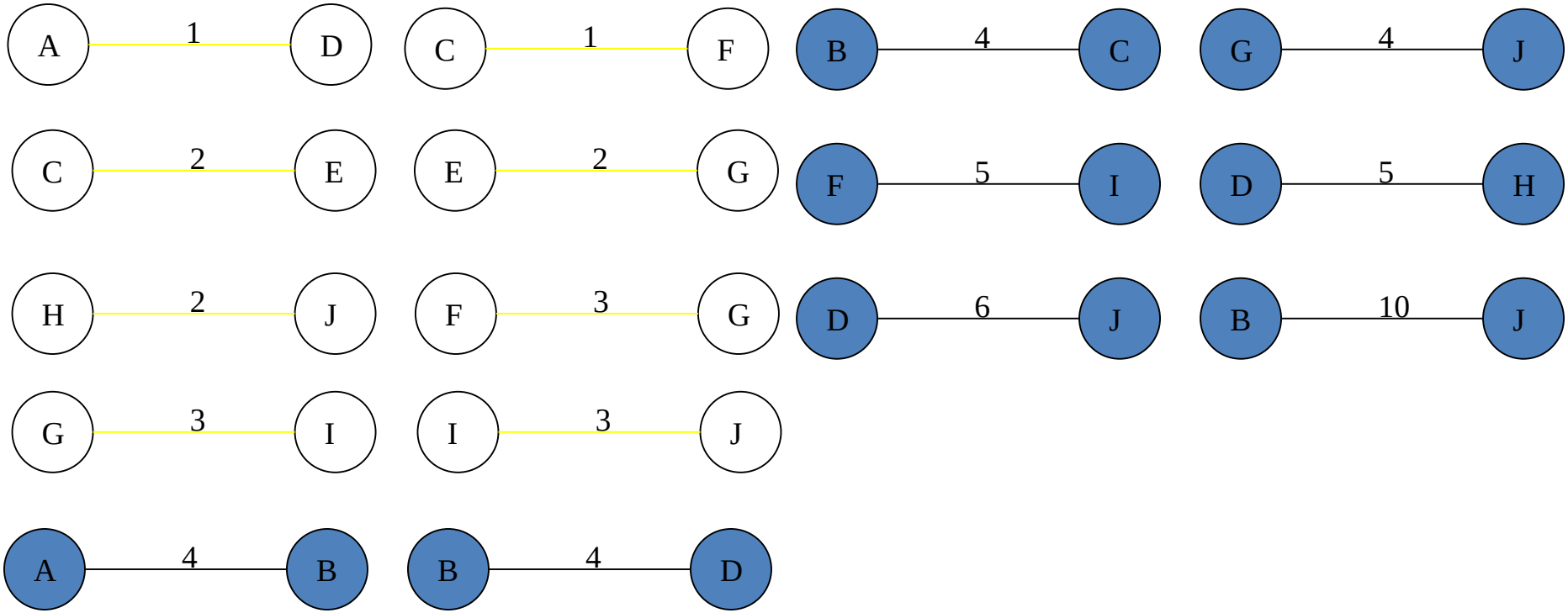
Αλγόριθμος Kruskal – Παράδειγμα XVI
Προσθήκη ακμής



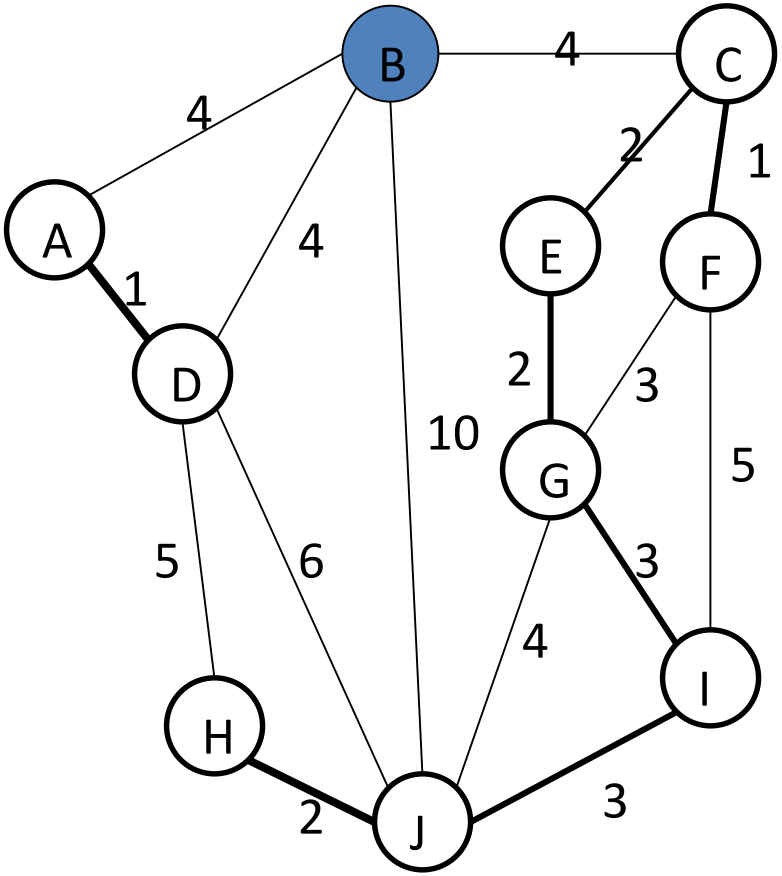
Αλγόριθμος Kruskal – Παράδειγμα XVII
Προσθήκη ακμής



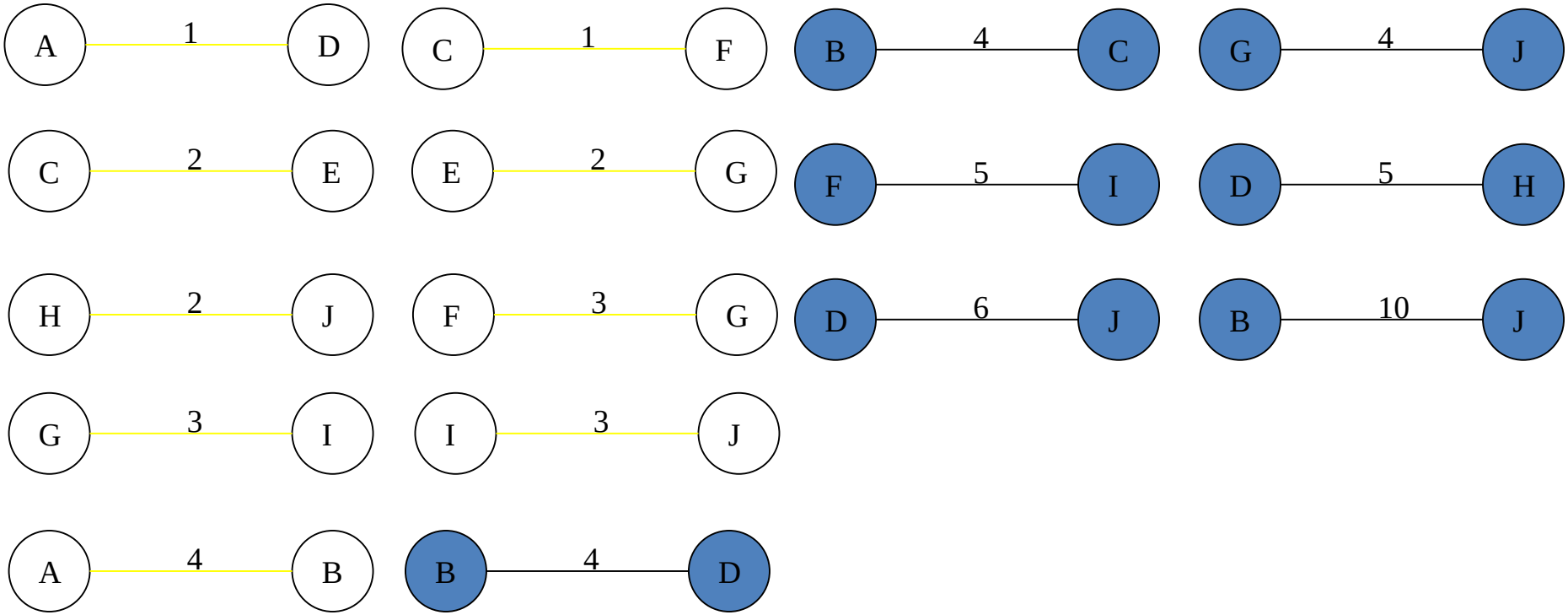
Αλγόριθμος Kruskal – Παράδειγμα XVIII
Προσθήκη ακμής



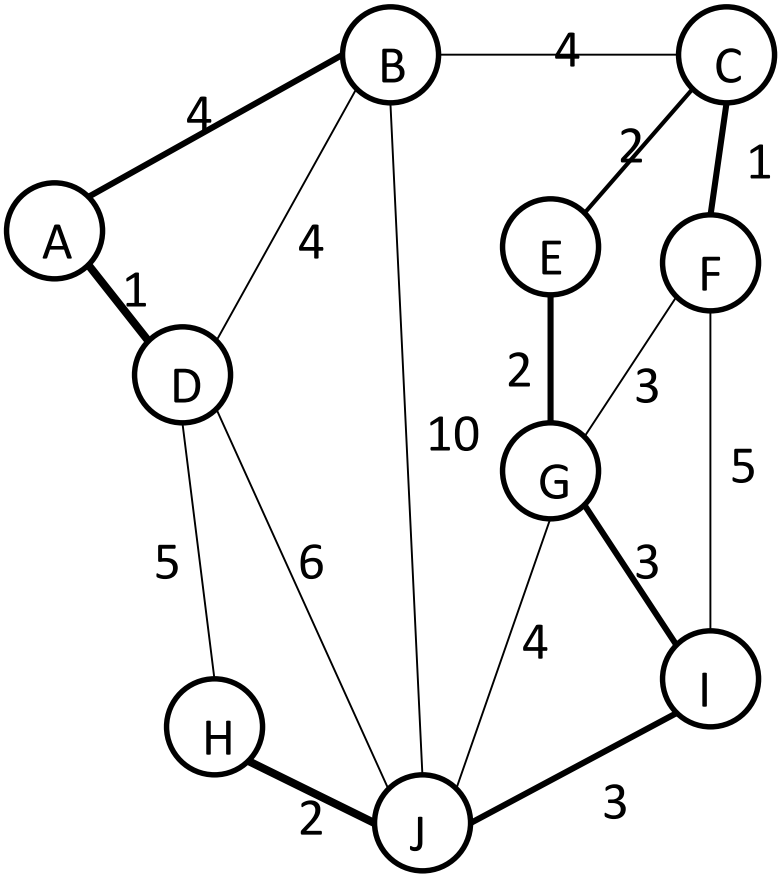
Αλγόριθμος Kruskal – Παράδειγμα XIX
Προσθήκη ακμής



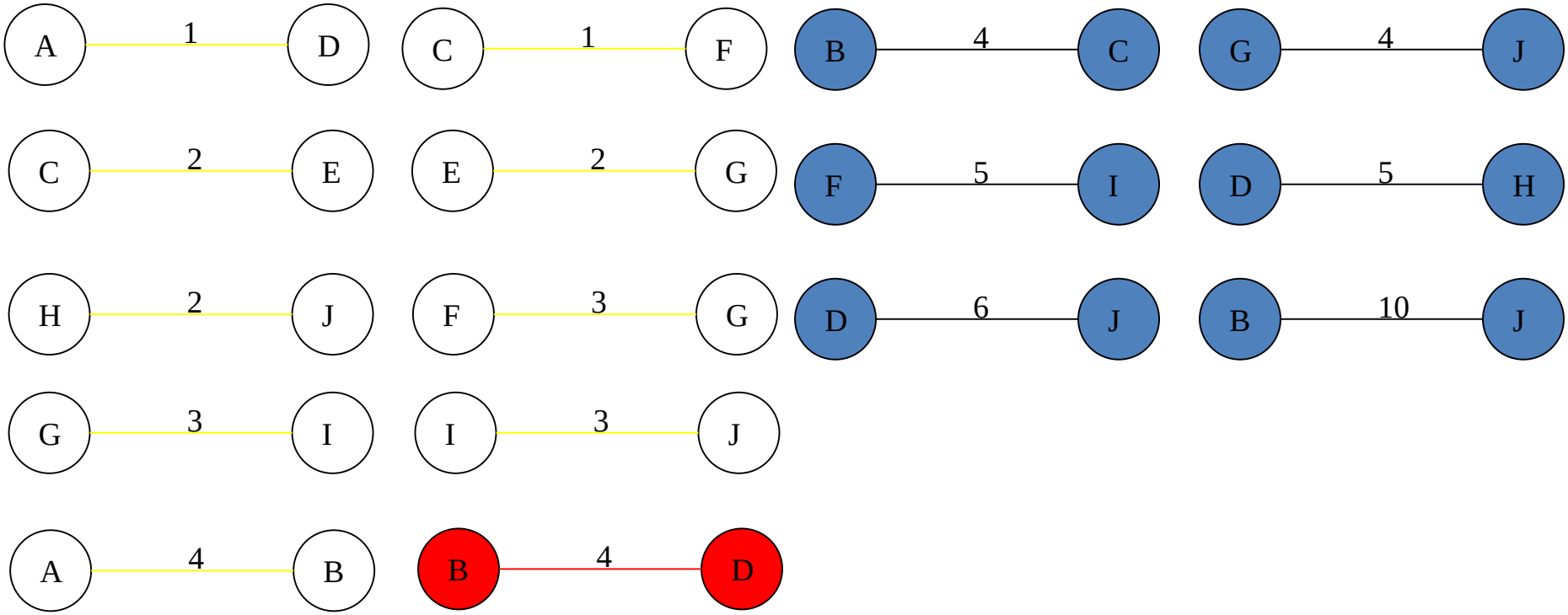
Αλγόριθμος Kruskal – Παράδειγμα ΧΧ Προσθήκη ακμής



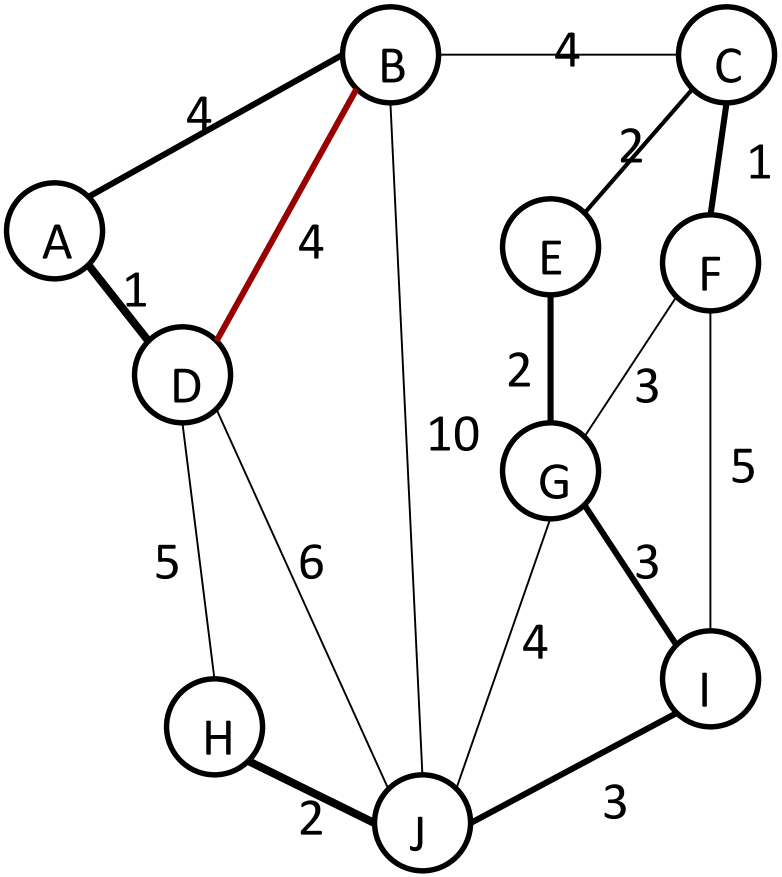
Αλγόριθμος Kruskal – Παράδειγμα XXI
Προσθήκη ακμής



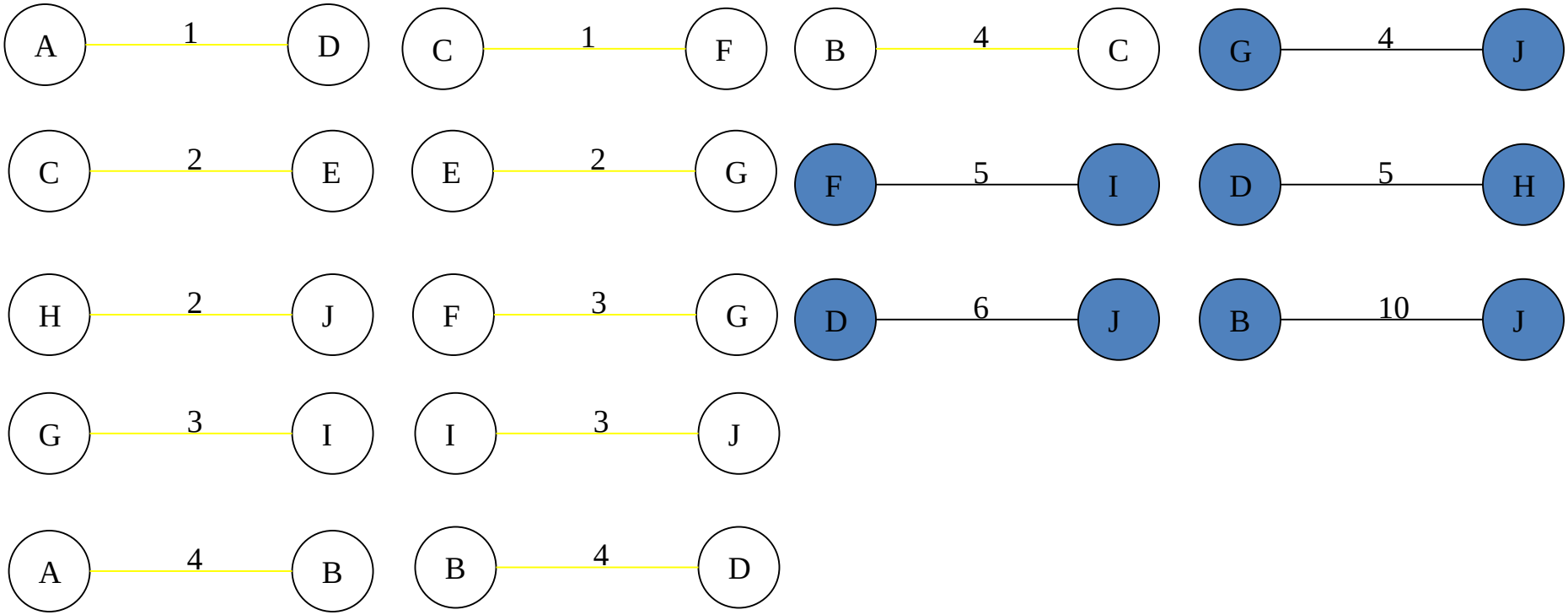
Αλγόριθμος Kruskal – Παράδειγμα XXII
Δημιουργία κύκλου



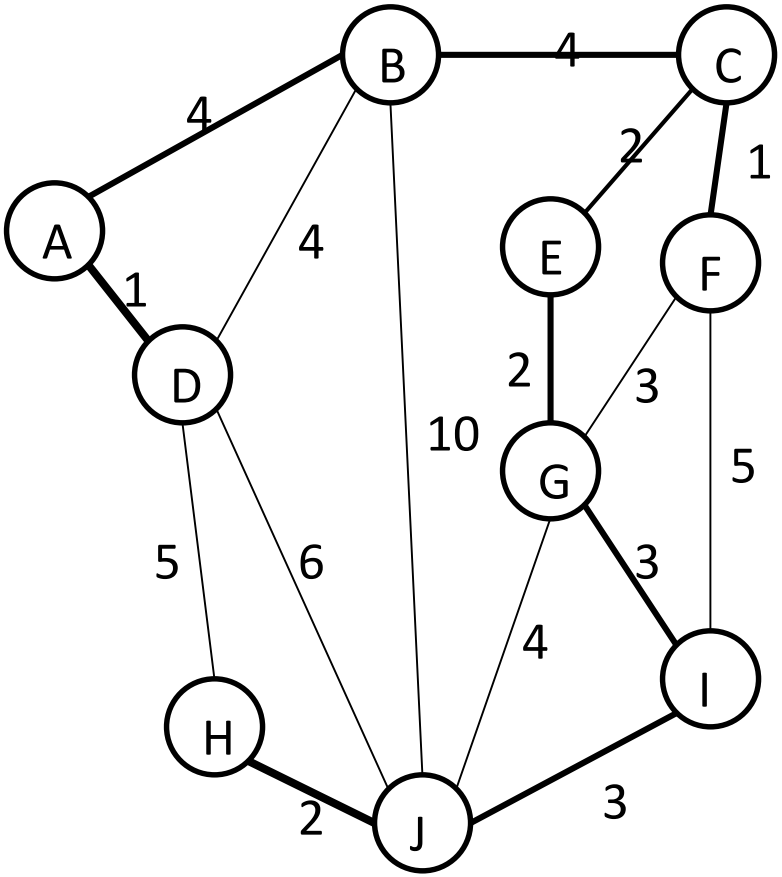
Αλγόριθμος Kruskal – Παράδειγμα XXIII
Δημιουργία κύκλου



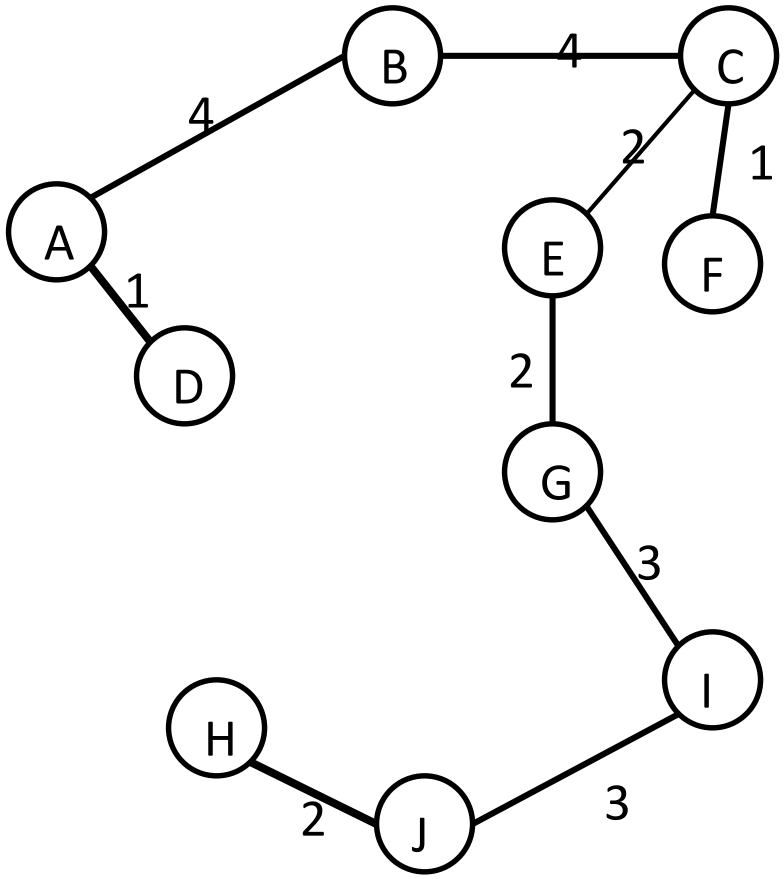
Αλγόριθμος Kruskal – Παράδειγμα XXIV
Προσθήκη ακμής



Αλγόριθμος Kruskal – Παράδειγμα XXV
Προσθήκη ακμής



Αλγόριθμος Kruskal – Παράδειγμα XXVI
Ελάχιστο Ζαευγνύον Δένδρο



Ανάλυση του Αλγορίθμου Kruskal

- Χρόνος Εκτέλεσης= O(m log n) (m = ακμές, n = κόμβοι)
- Ο έλεγχος αν μία ακμή δημιουργεί κύκλο μπορεί να είναι «αργός», εκτός και αν χρησιμοποιηθεί μία πολύ-πλοκη δομή δεδομένων, όπως η δομή “union-find”.
- Συνήθως χρειάζεται να ελέγξουμε ένα μικρό μέρος των ακμών, αλλά σε ορισμένες περιπτώσεις (όπως αν υπάρχει κόμβος ο οποίος συνδέεται με μία μόνο ακμή στο γράφο η οποία είναι η πιο ακριβή) χρειάζεται να ελέγξουμε όλες τις ακμές.
- Ο αλγόριθμος λειτουργεί καλύτερα εάν ο αριθμός των ακμών διατηρείται στο ελάχιστο.

Αλγόριθμος Prim I

- Αυτός ξεκινά με ένα κόμβο.
- Στη συνέχεια προσθέτει ένα κόμβο κάθε φορά που δεν είναι συνδεδεμένος στο νέο γράφο, επιλέγοντας τον κόμβο με προσπίπτουσες ακμές που έχουν το ελάχιστο βάρος.

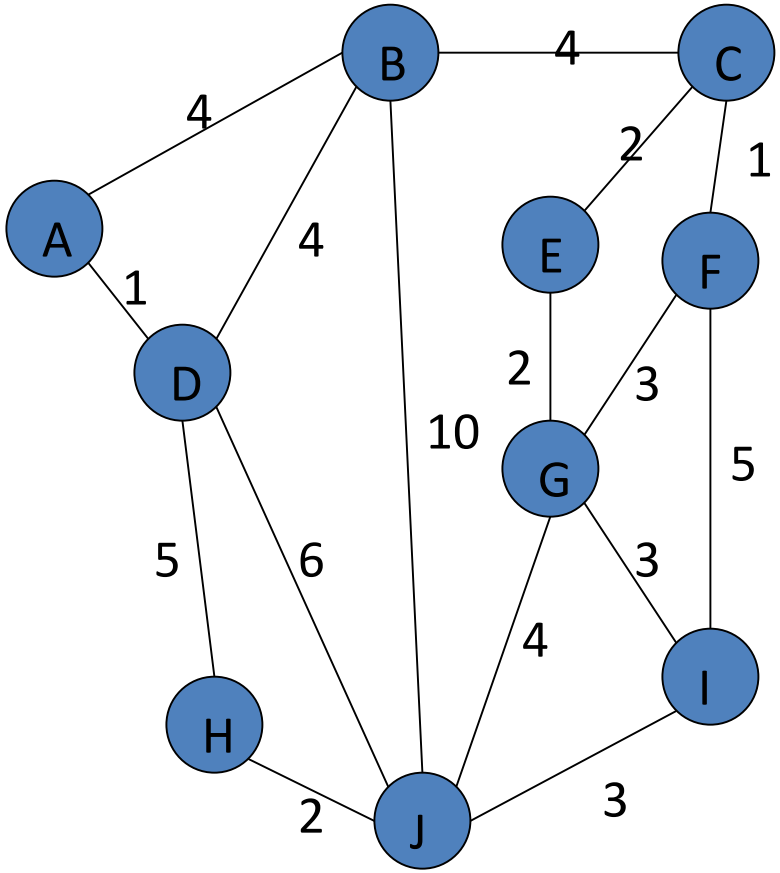
Αλγόριθμος Prim II

Τα βήματα είναι:

1. Κατασκευάζεται ο νέος γράφος με ένα κόμβο από τον παλιό γράφο.
2. Καθώς ο νέος γράφος έχει λιγότερους από n κόμβους,
    1. Βρες έναν κόμβο από τον παλιό γράφο με τη «μικρότερη» ακμή, η οποία συνδέει τον κόμβο με το νέο γράφο,
    2. Πρόσθεσε τον κόμβο στο νέο γράφο

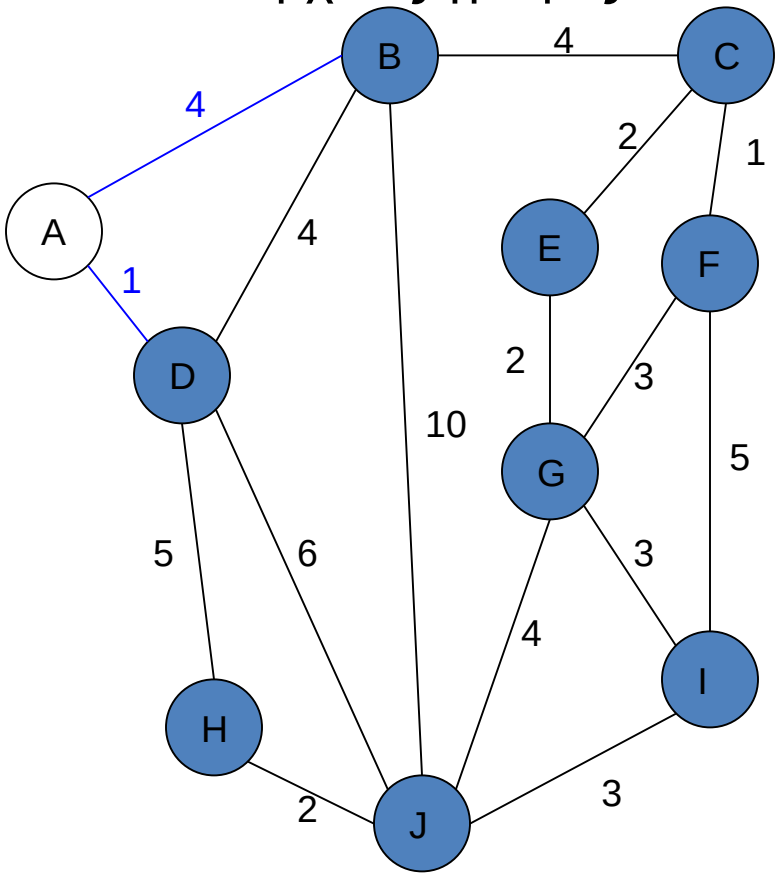
- Σε κάθε βήμα ενώνουμε ένα κόμβο, επομένως στο τέλος θα έχουμε ένα γράφο με όλους τους κόμβους και θα είναι το ελάχιστο ζευγνύον δένδρο του αρχικού γράφου.

Αλγόριθμος Prim – Παράδειγμα Ι

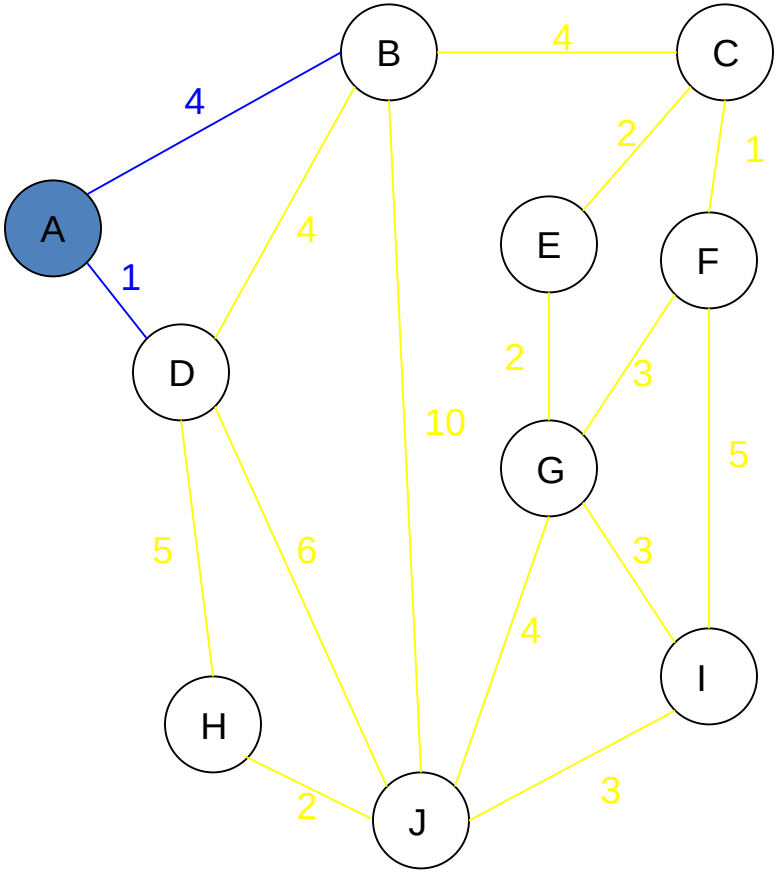


Αλγόριθμος Prim – Παράδειγμα ΙΙ

Αρχικός γράφος

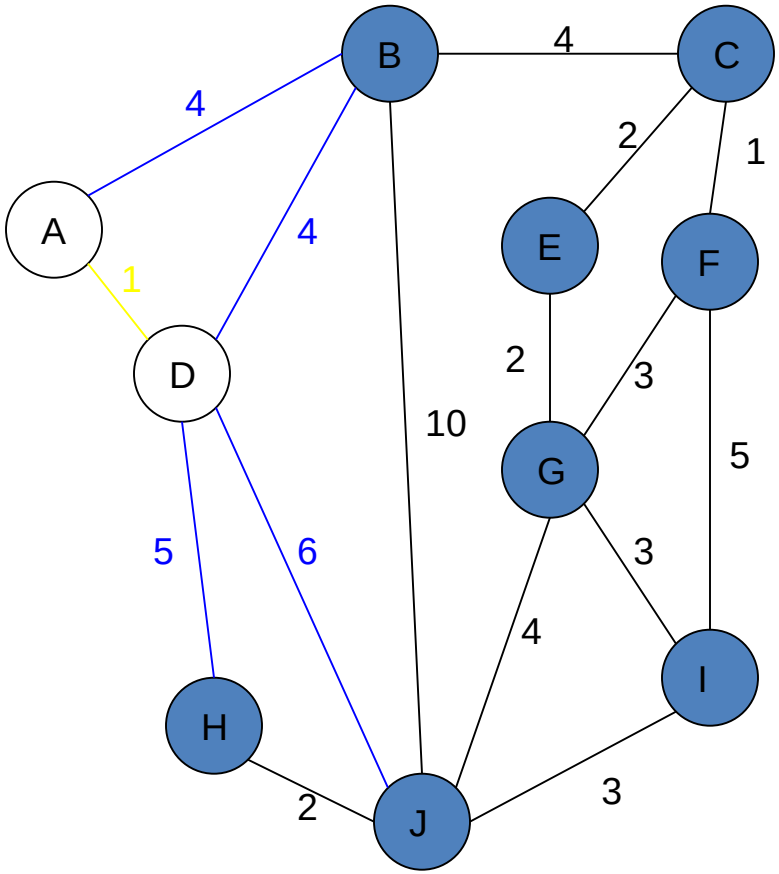


Νέος γράφος

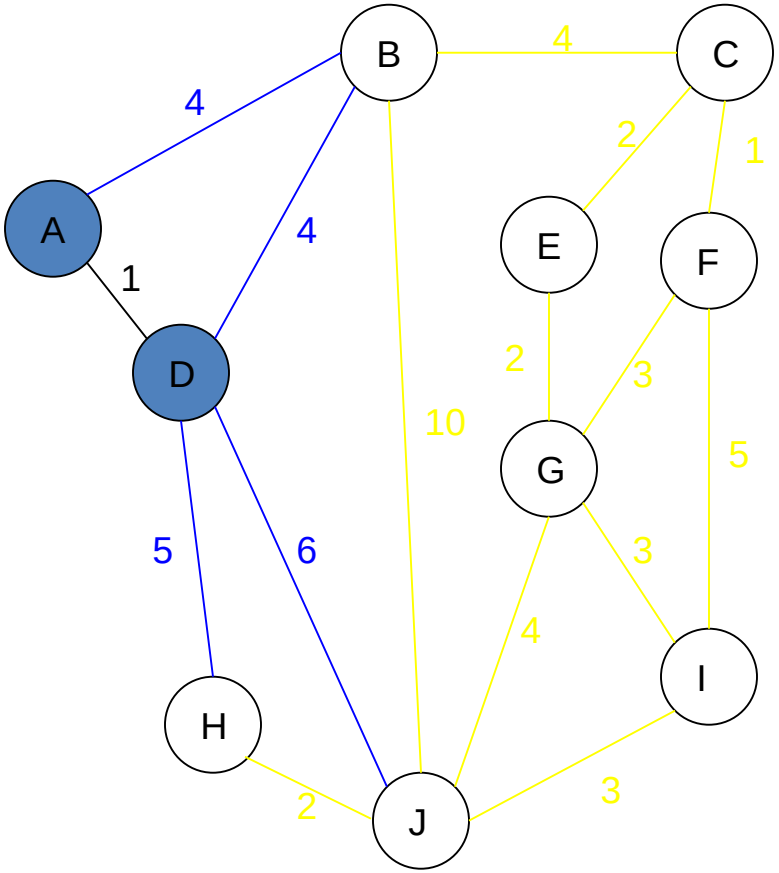


Αλγόριθμος Prim – Παράδειγμα III

Αρχικός γράφος

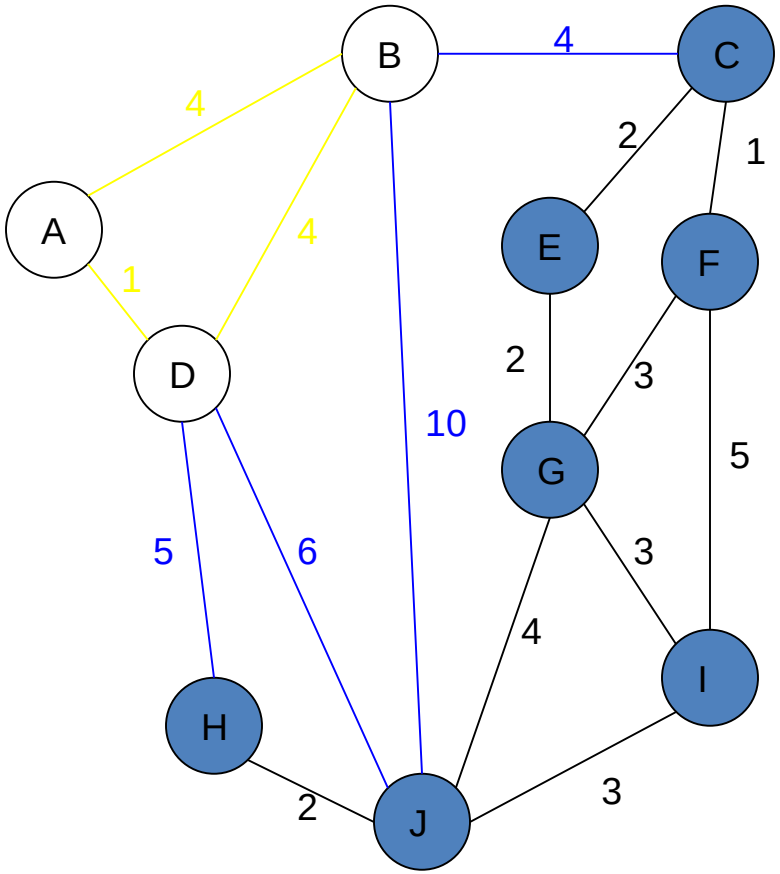


Νέος γράφος

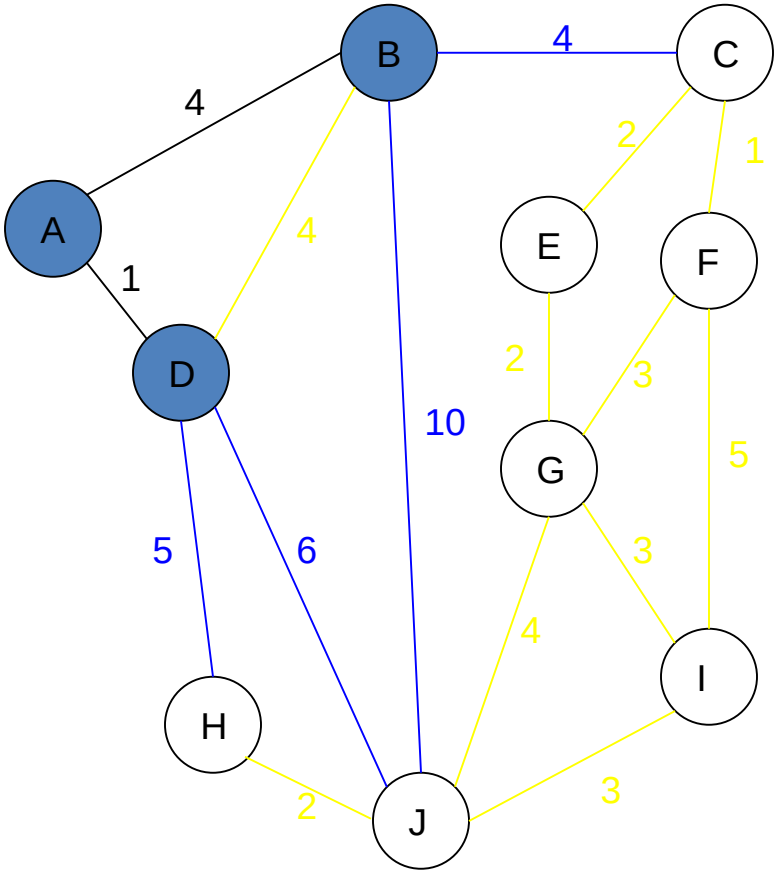


Αλγόριθμος Prim – Παράδειγμα IV

Αρχικός γράφος

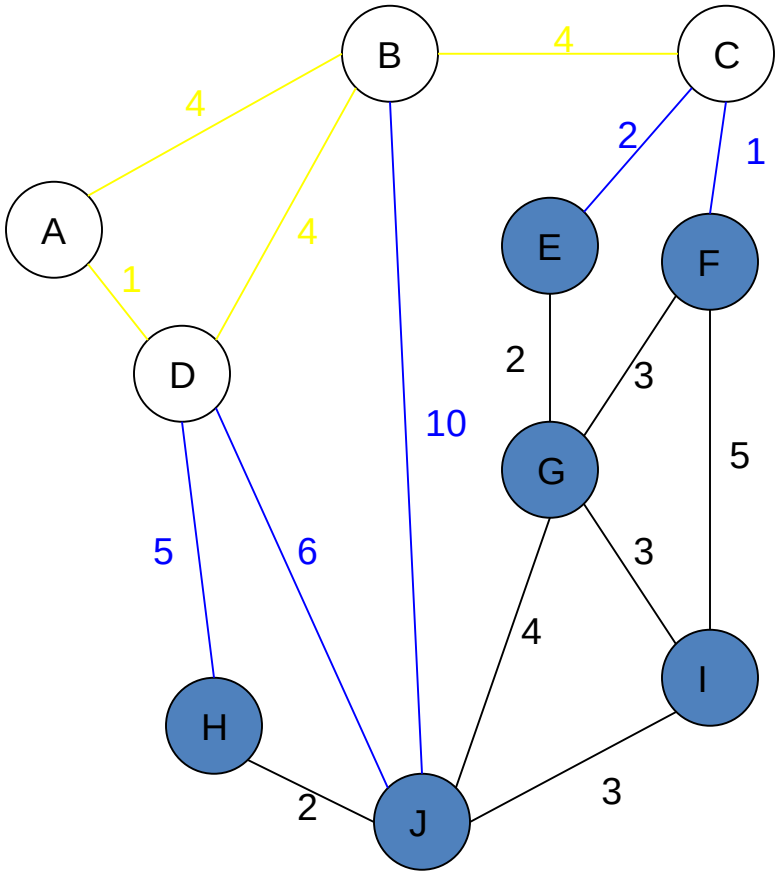


Νέος γράφος

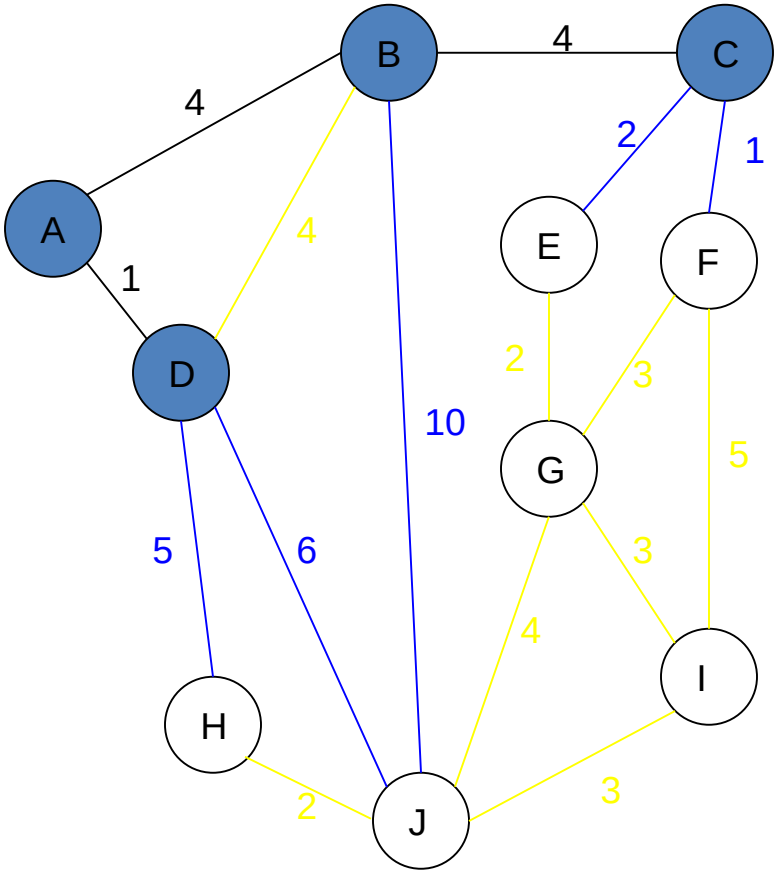


Αλγόριθμος Prim – Παράδειγμα V

Αρχικός γράφος

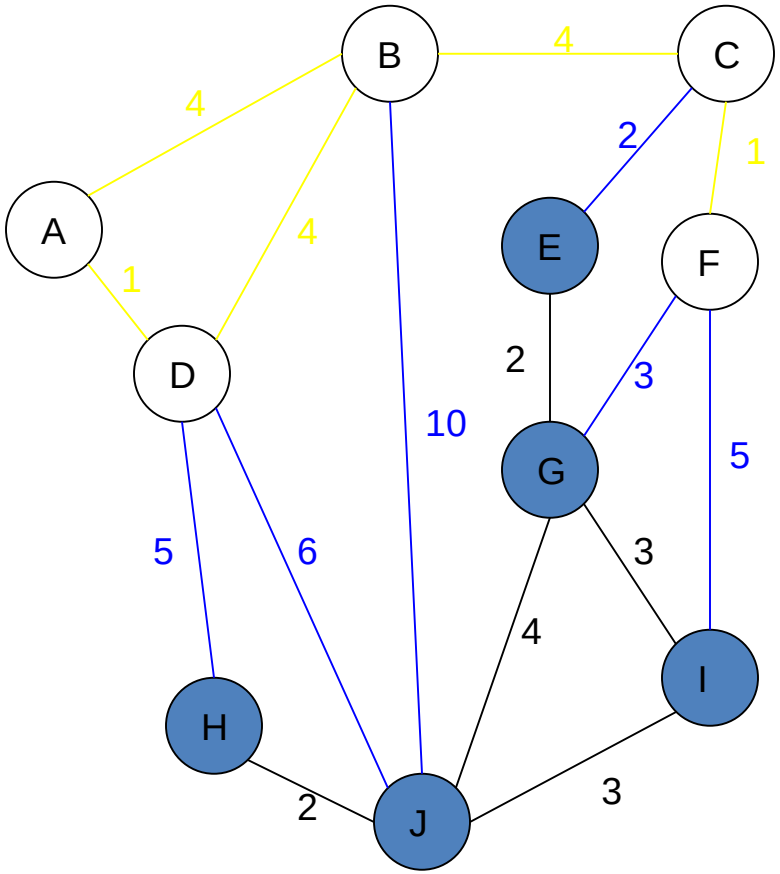


Νέος γράφος

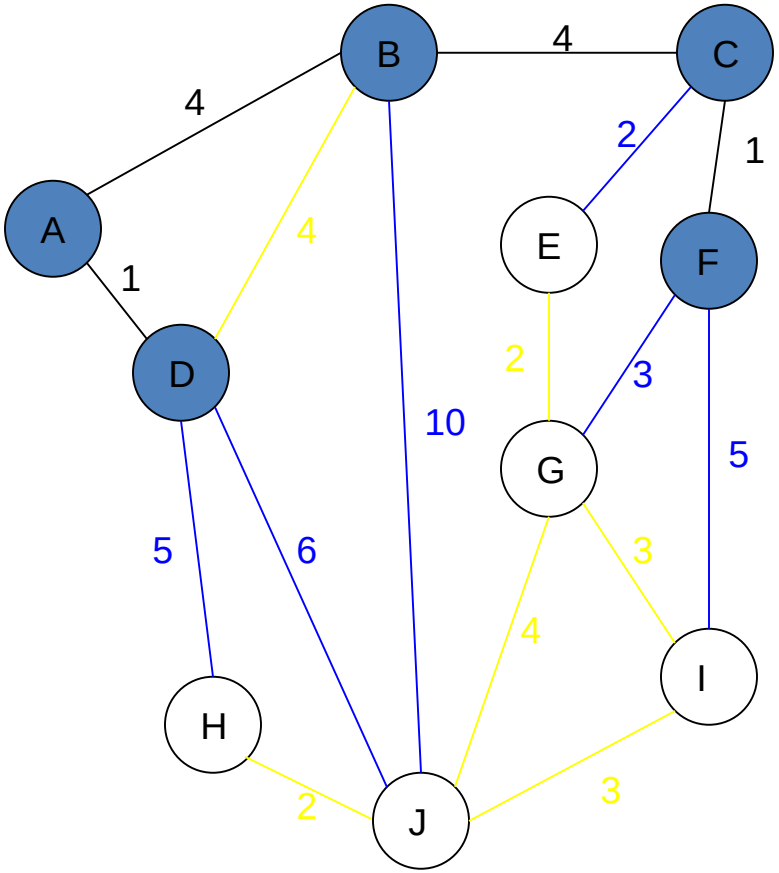


Αλγόριθμος Prim – Παράδειγμα VI

Αρχικός γράφος

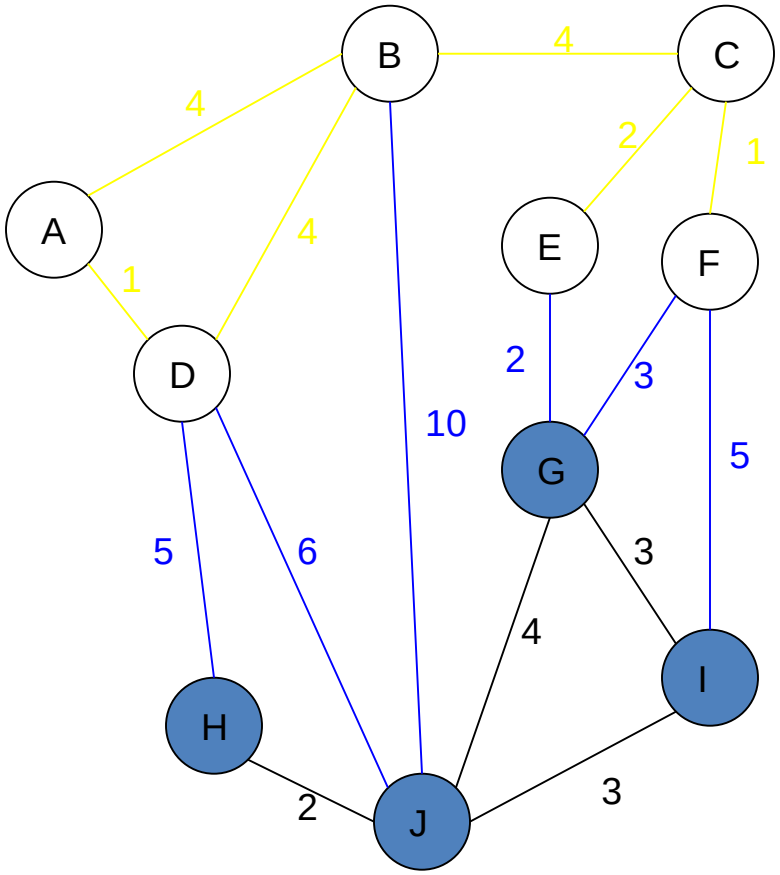


Νέος γράφος

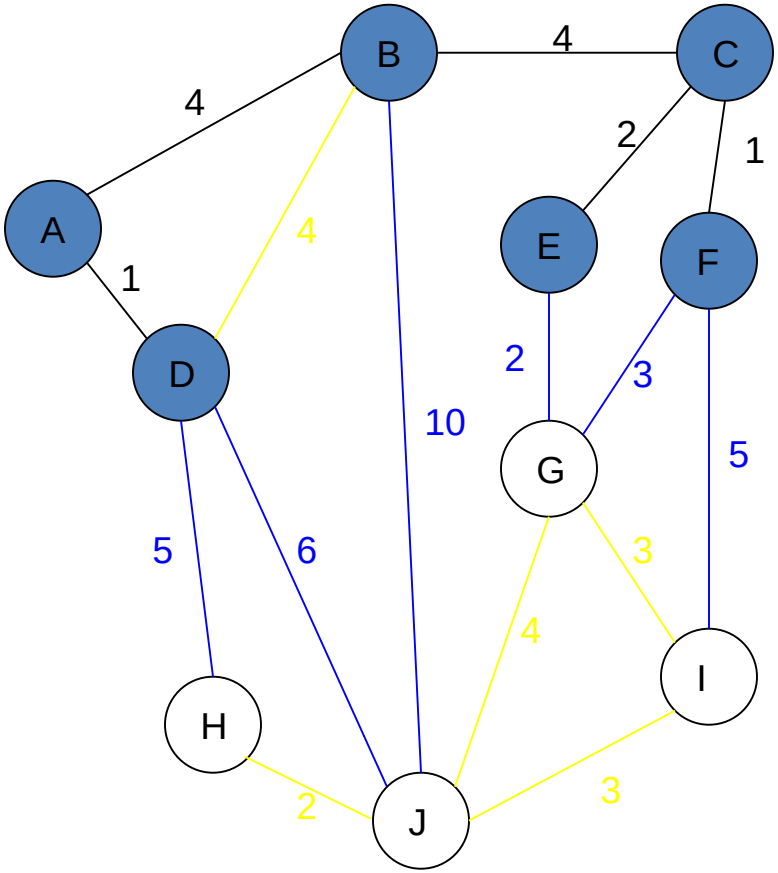


Αλγόριθμος Prim – Παράδειγμα VII

Αρχικός γράφος

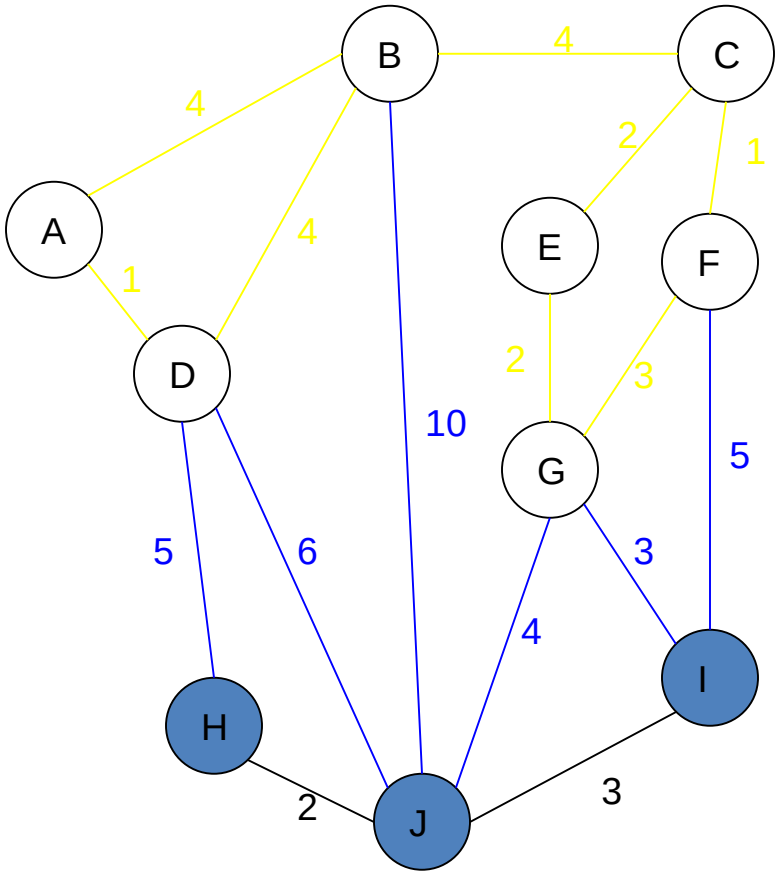


Νέος γράφος

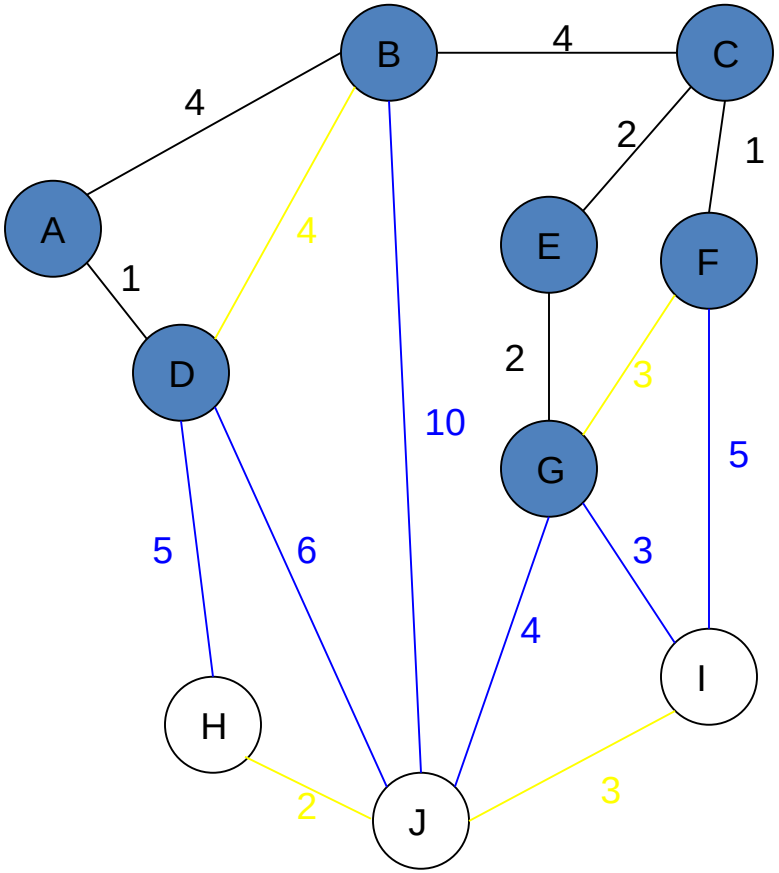


Αλγόριθμος Prim – Παράδειγμα VIII

Αρχικός γράφος

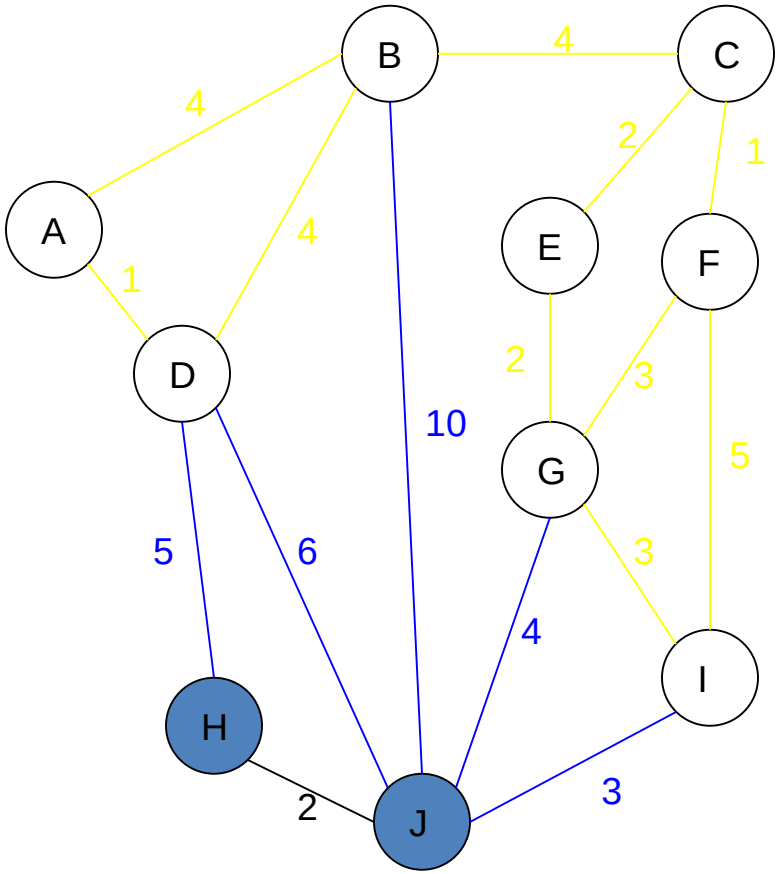


Νέος γράφος

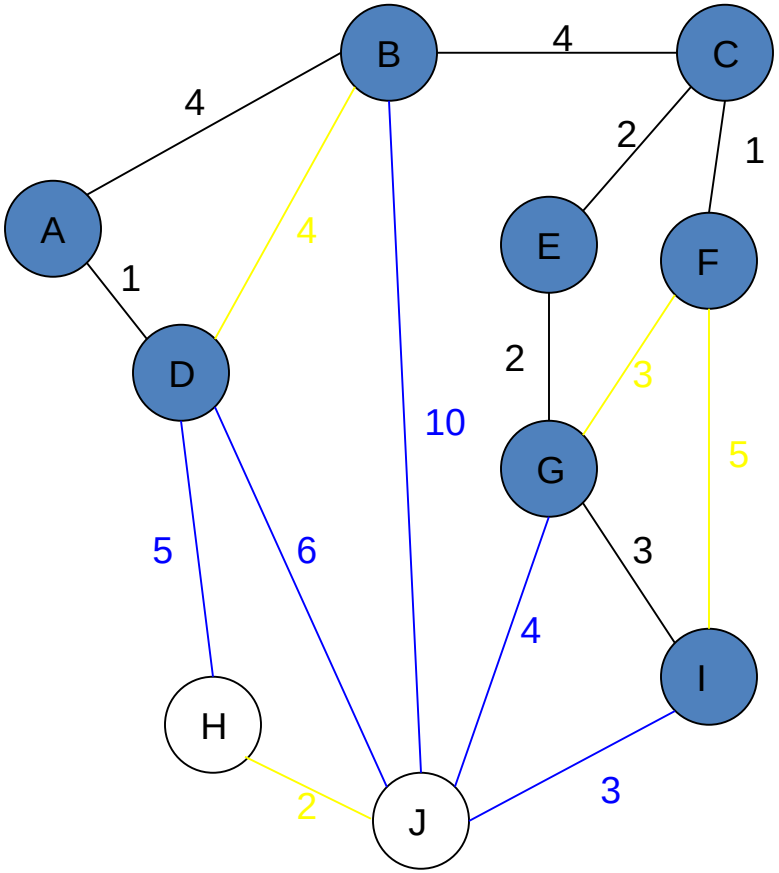


Αλγόριθμος Prim – Παράδειγμα IX

Αρχικός γράφος

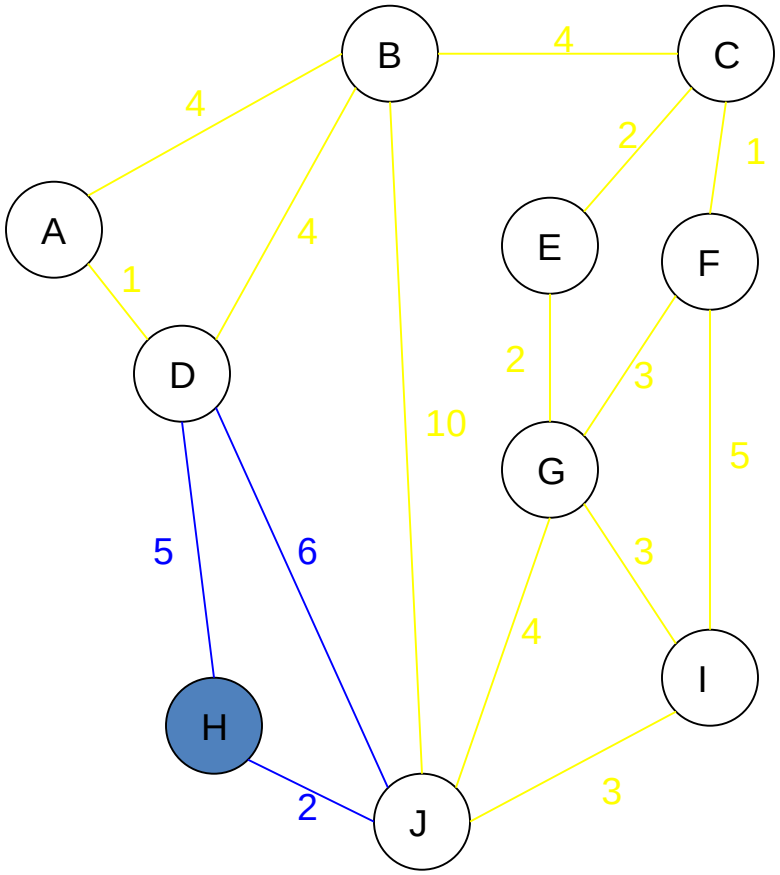


Νέος γράφος

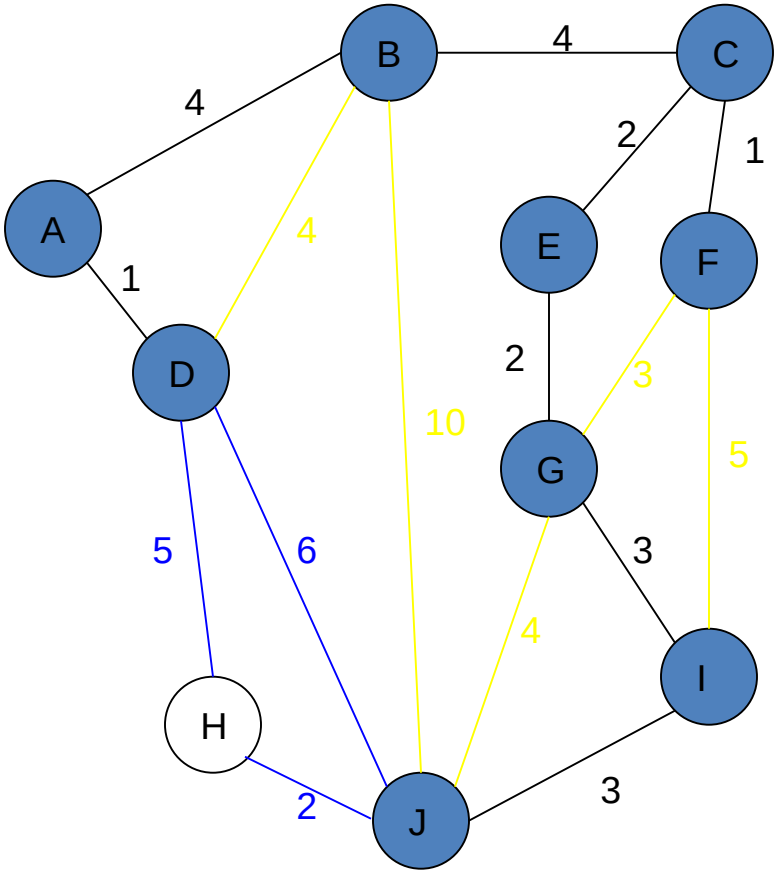


Αλγόριθμος Prim – Παράδειγμα Χ

Αρχικός γράφος

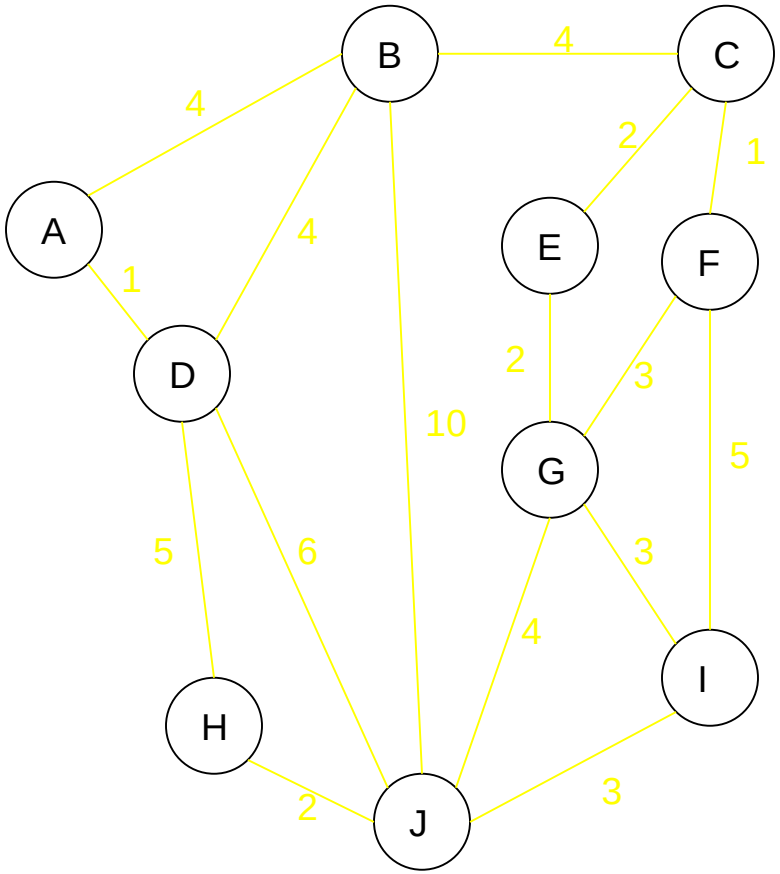


Νέος γράφος

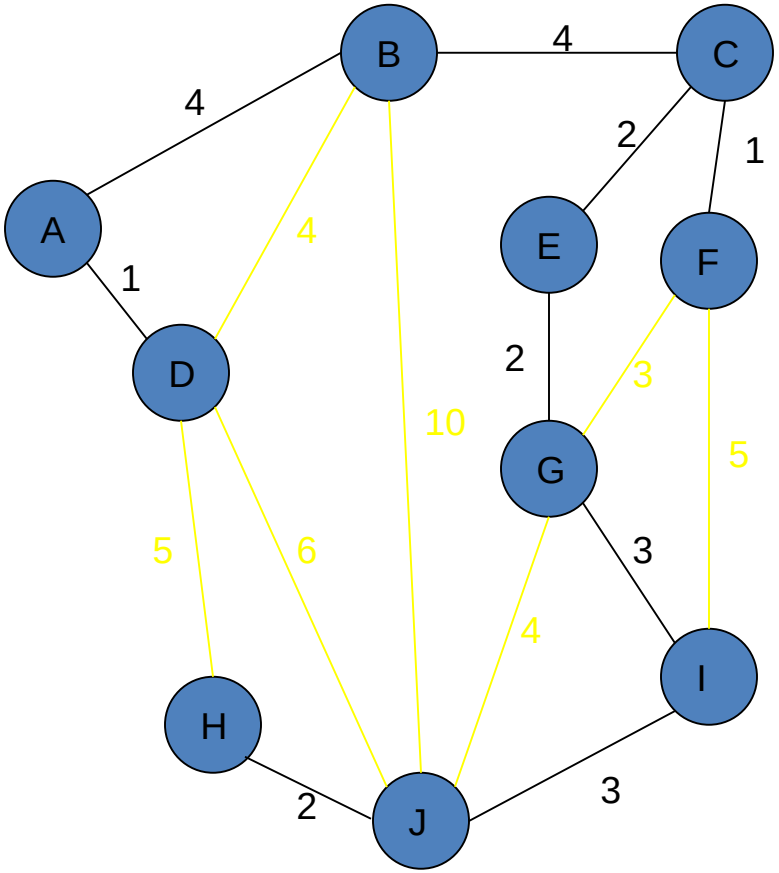


Αλγόριθμος Prim – Παράδειγμα XI

Αρχικός γράφος

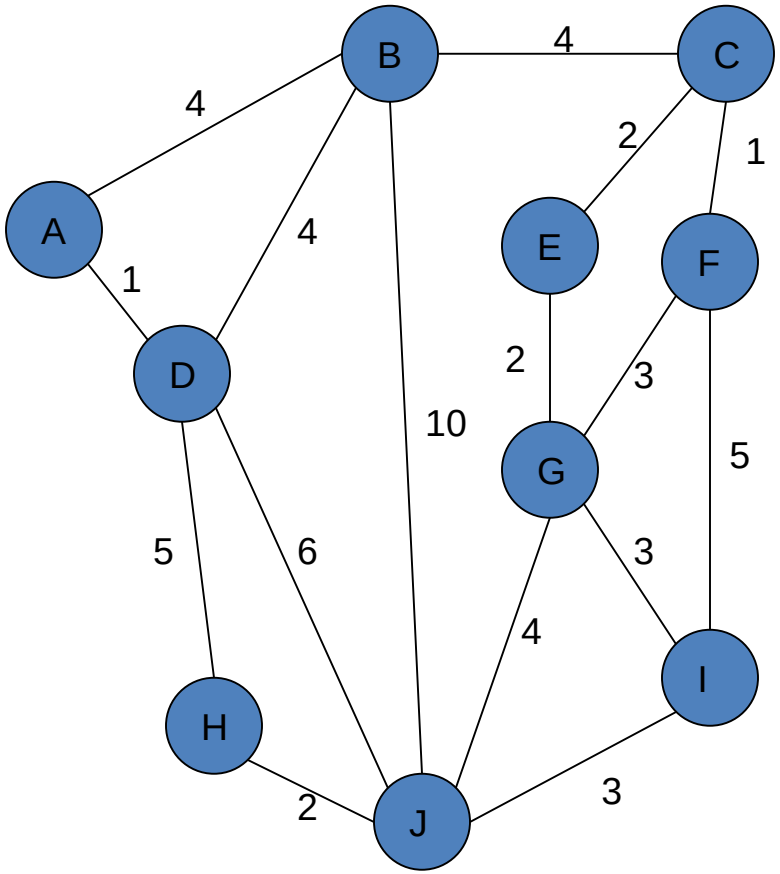


Νέος γράφος

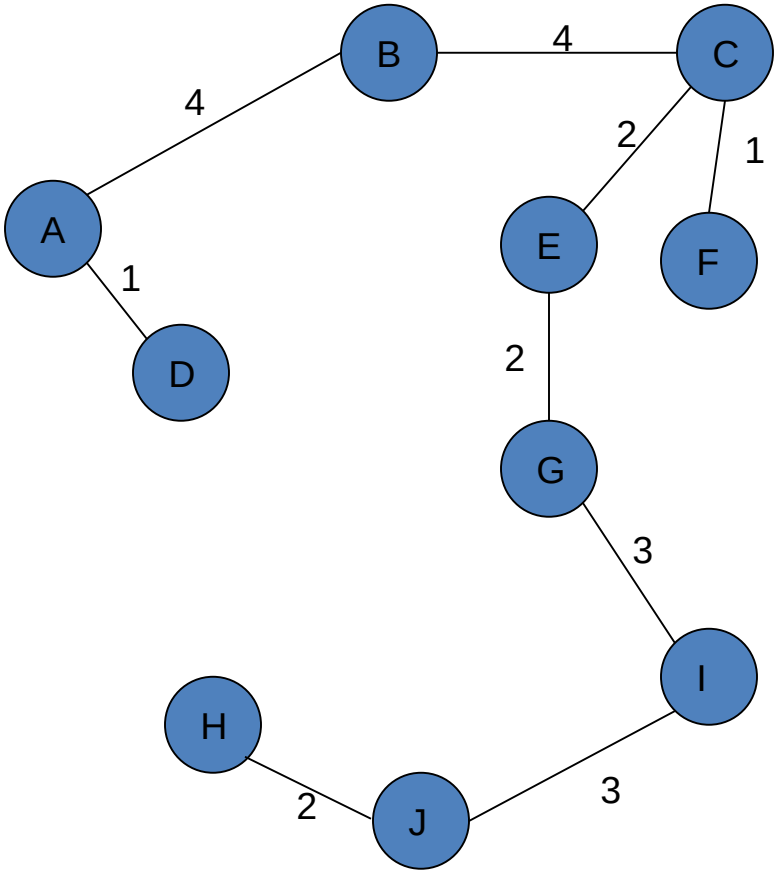


Αλγόριθμος Prim – Παράδειγμα XII

Πλήρης γράφος



Ελάχιστο ζευγνύον δένδρο



Ανάλυση του Αλγορίθμου Prim

- Χρόνος Εκτέλεσης = O(m + n log n) (m=ακμές, n=κορυφές)
- Εάν δεν χρησιμοποιηθεί σωρός, ο χρόνος εκτέλεσης θα είναι O(n^2) αντί για O(m + n log n). Όμως η χρήση σωρού περιπλέκει τον κώδικα δεδομένου ότι περιπλέκεται η δομή δεδομένων. Ένας σωρός Fibonacci είναι ο καλύτερος σωρός προς χρήση, αλλά και πάλι περιπλέκει τον κώδικα.
- Αντίθετα με τον αλγόριθμο του Kruskal, δεν χρειάζεται να εξετάσει το γράφο απευθείας. Μπορεί να ασχολείται με ένα κομμάτι κάθε φορά. Δεν χρειάζεται επίσης να απασχολείται με το γεγονός αν η προσθήκη μίας ακμής θα δημιουργήσει κύκλο, αφού ο αλγόριθμος ασχολείται κυρίως ε τους κόμβους και όχι με τις ακμές.
- Για το συγκεκριμένο αλγόριθμος το πλήθος των κόμβων πρέπει να διατηρείται στο ελάχιστο, όπως και το πλήθος των ακμών. Για μικρούς γράφους, το πλήθος των ακμών έχει μεγαλύτερη σημασία, ενώ για μεγάλους γράφους έχει μεγαλύτερη σημασία το πλήθος των κόμβων.

Αλγόριθμος Boruvka I

- Είναι παρόμοιος με τον αλγόριθμο του Prim, αλλά οι κόμβοι προστίθενται στον νέο γράφο παράλληλα σε ολόκληρο το γράφο.
- Δημιουργεί μία λίστα από δένδρα, κάθε ένα από τα οποία αποτελείται από ένα κόμβο του αρχικού γράφου και προχωρά στη συγχώνευση τους με τις «μικρότερες» ακμές, μέχρι να υπάρξει ένα μόνο δένδρο, το οποίο είναι και το ελάχιστο ζευγνύον δένδρο.
- Δουλεύει παρόμοια με τον merge sort.

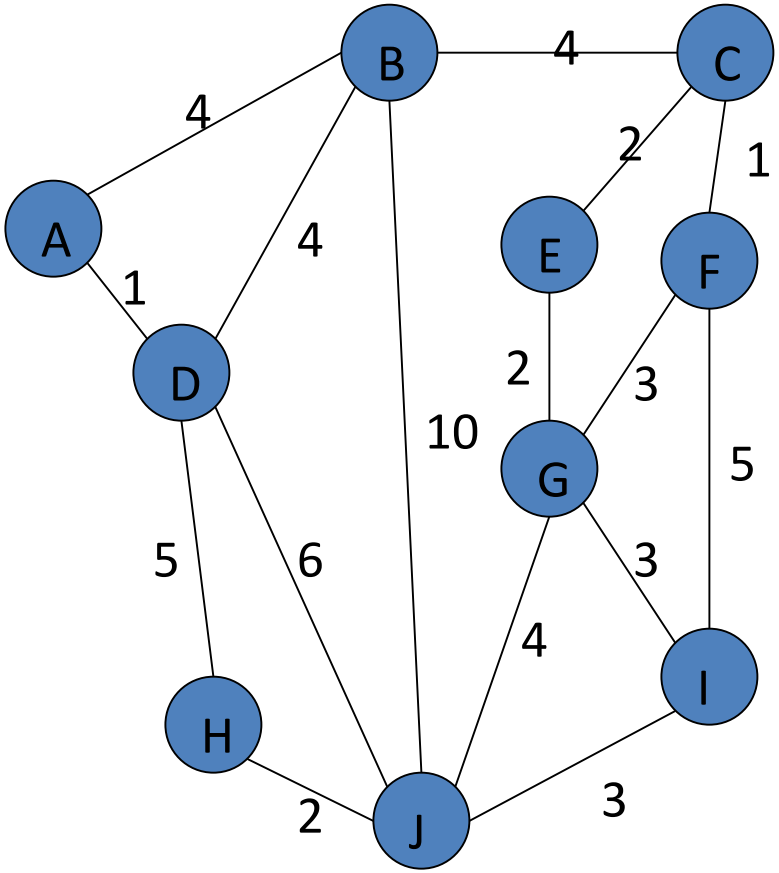
Αλγόριθμος Boruvka II

Τα βήματα είναι:

1. Δημιούργησε μία λίστα από n δένδρα που αποτελούνται από ένα κόμβο.
2. Όσο η λίστα περιέχει περισσότερο από ένα δένδρο,
    1. Για κάθε δένδρο στη λίστα, βρες τον κόμβο που δεν συνδέεται με το δένδρο, με τη «μικρότερη» συνδέουσα ακμή στο δένδρο,
    2. Προσέθεσε όλες τις ακμές που βρέθηκαν στο νέο γράφο, δημιουργώντας ένα νέο σύνολο από δένδρα

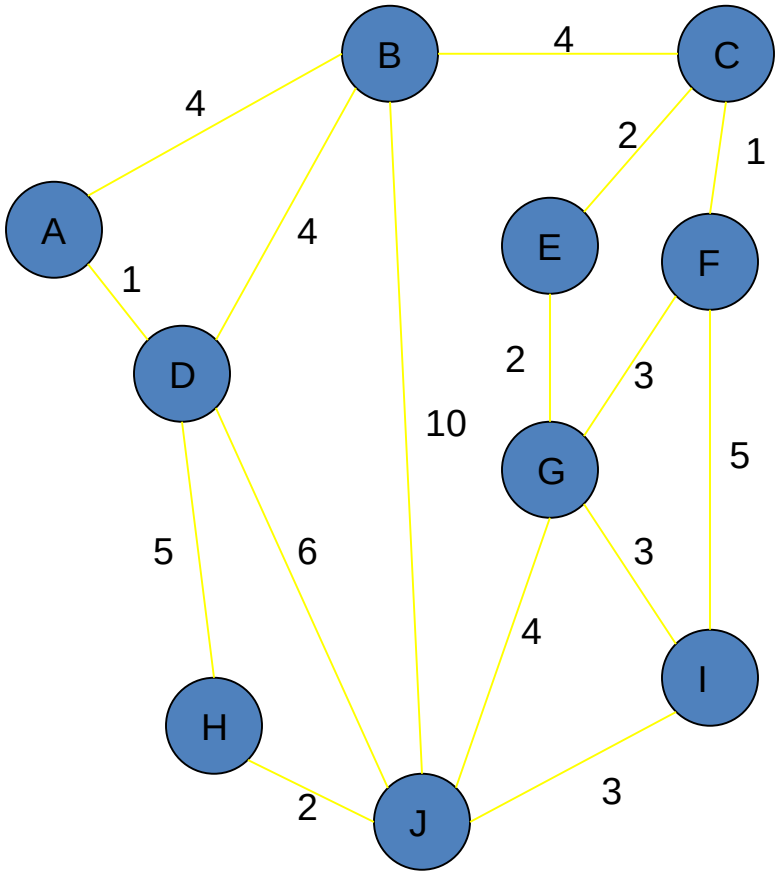
- Κάθε βήμα ενώνει γκρουπ από δένδρα μέχρι να μένει ένα δένδρο.

Αλγόριθμος Boruvka – Παράδειγμα Ι



Αλγόριθμος Boruvka – Παράδειγμα II

Αρχικά δένδρα του γράφου

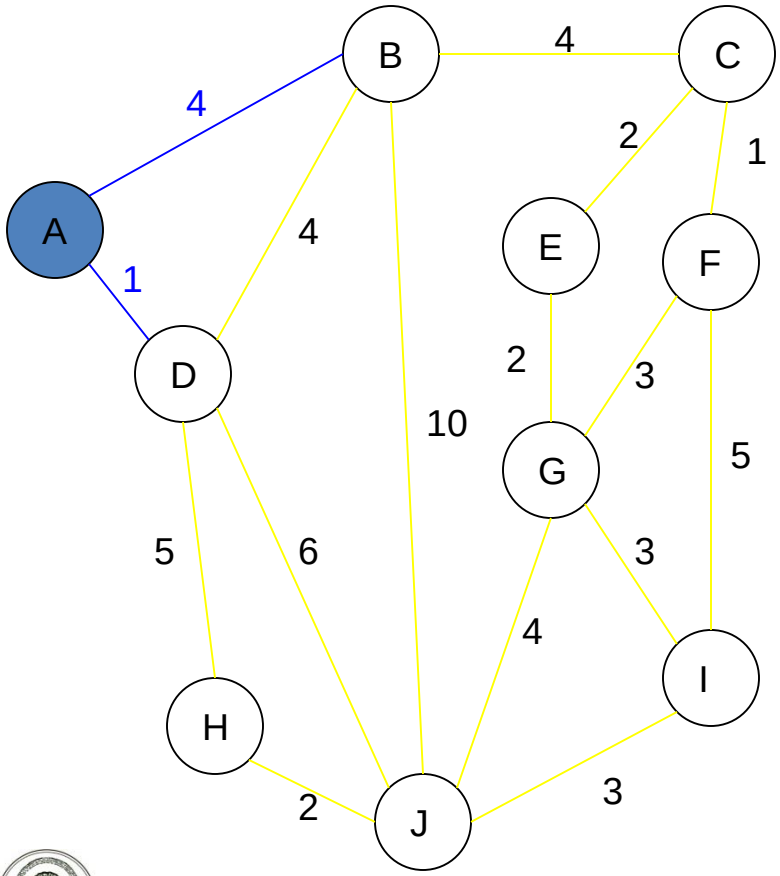


Λίστα δένδρων

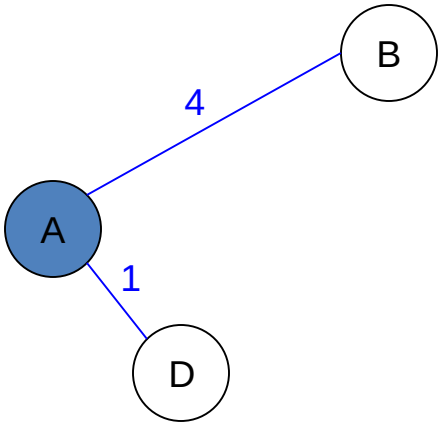
- A
- B
- C
- D
- E
- F
- G
- H
- I
- J

Αλγόριθμος Boruvka – Παράδειγμα III

1ος γύρος

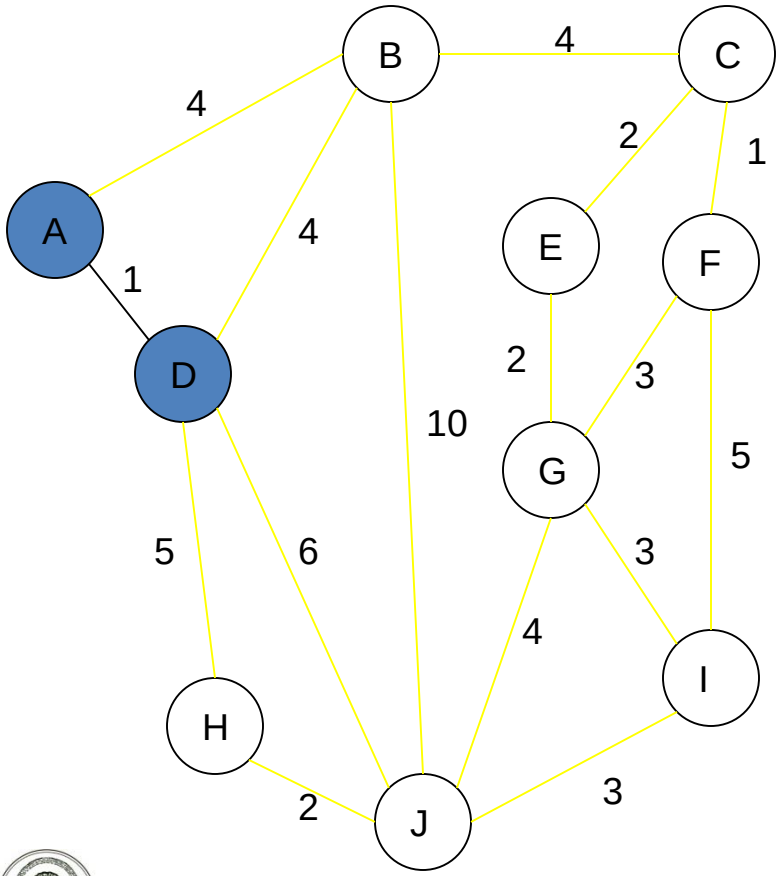


Δένδρο A

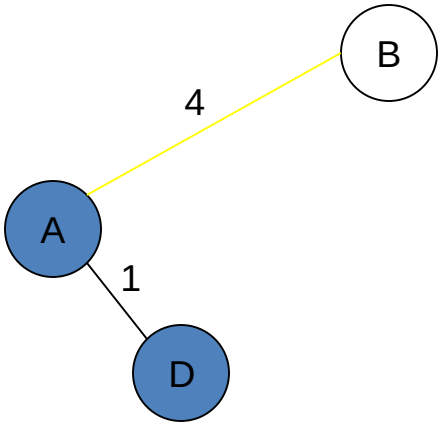


Αλγόριθμος Boruvka – Παράδειγμα IV

1ος γύρος

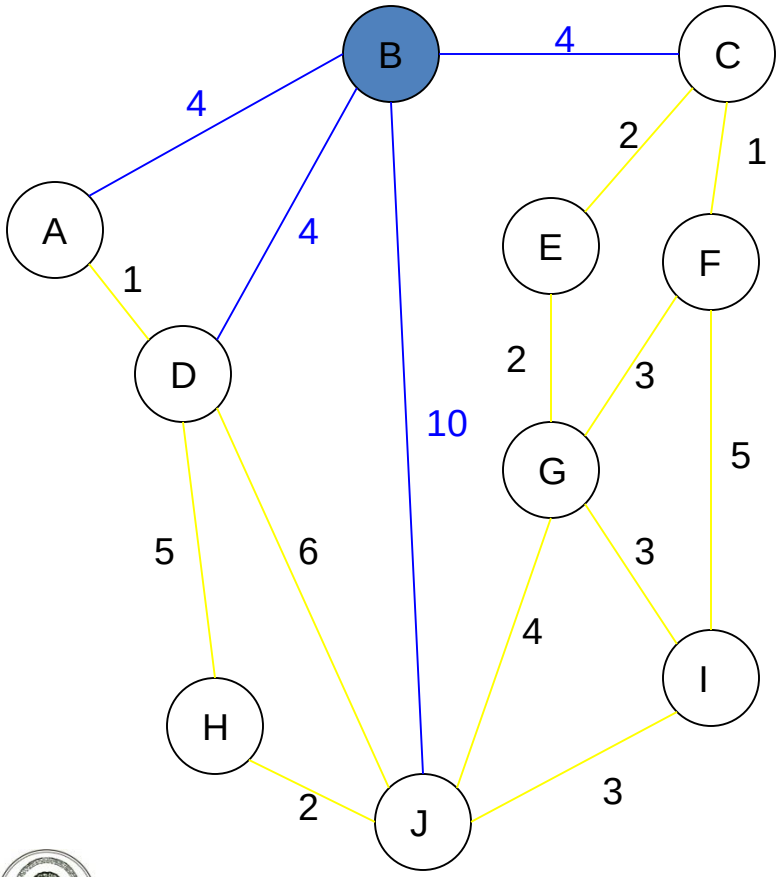


Ακμή A-D

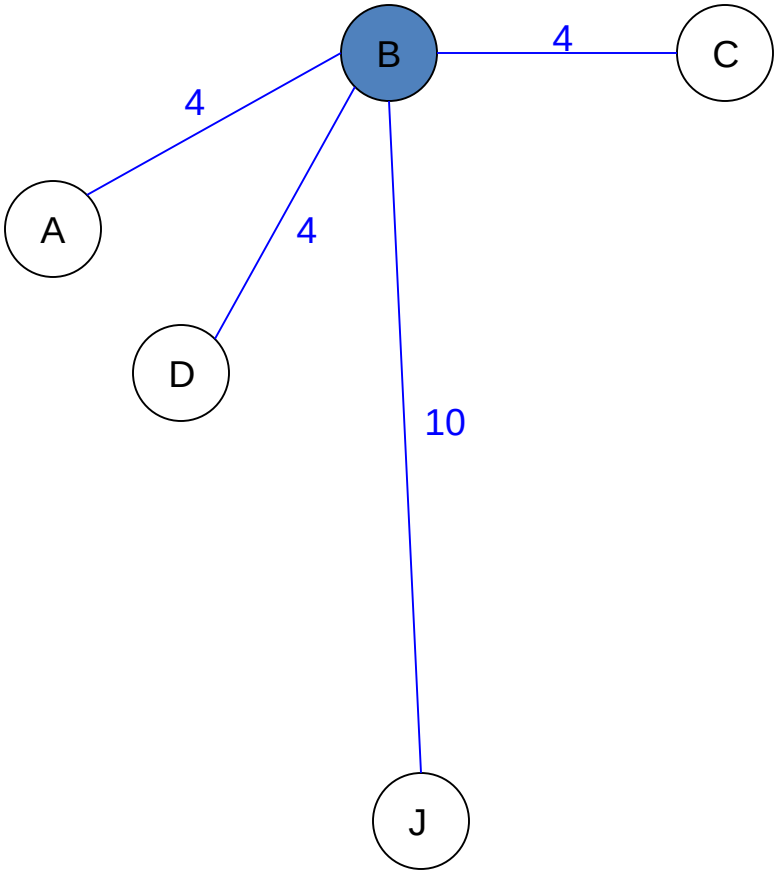


Αλγόριθμος Boruvka – Παράδειγμα V

1ος γύρος

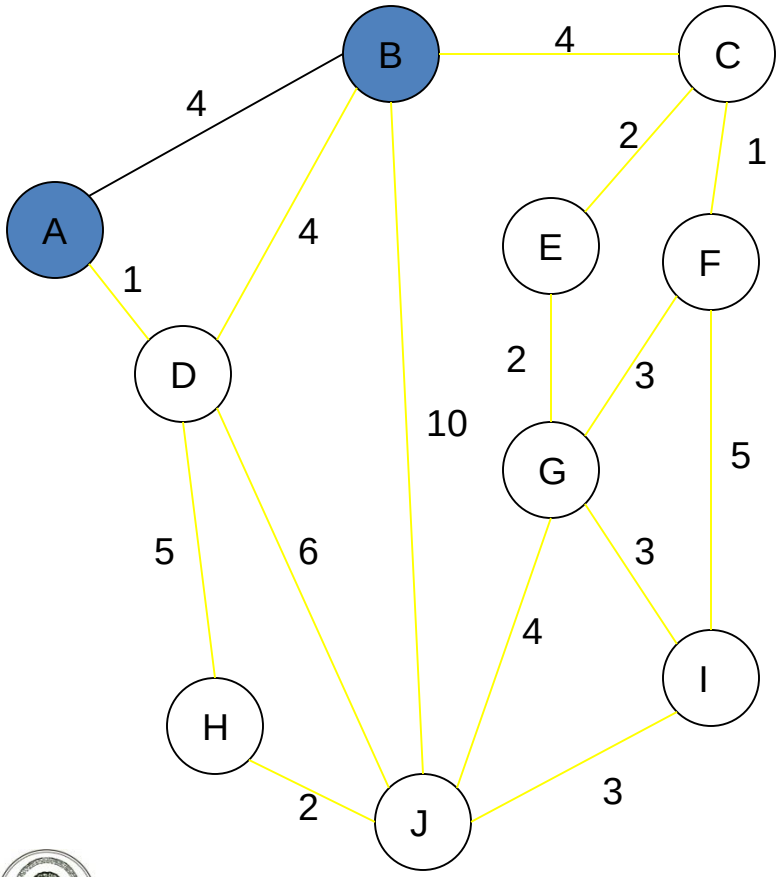


Δένδρο B

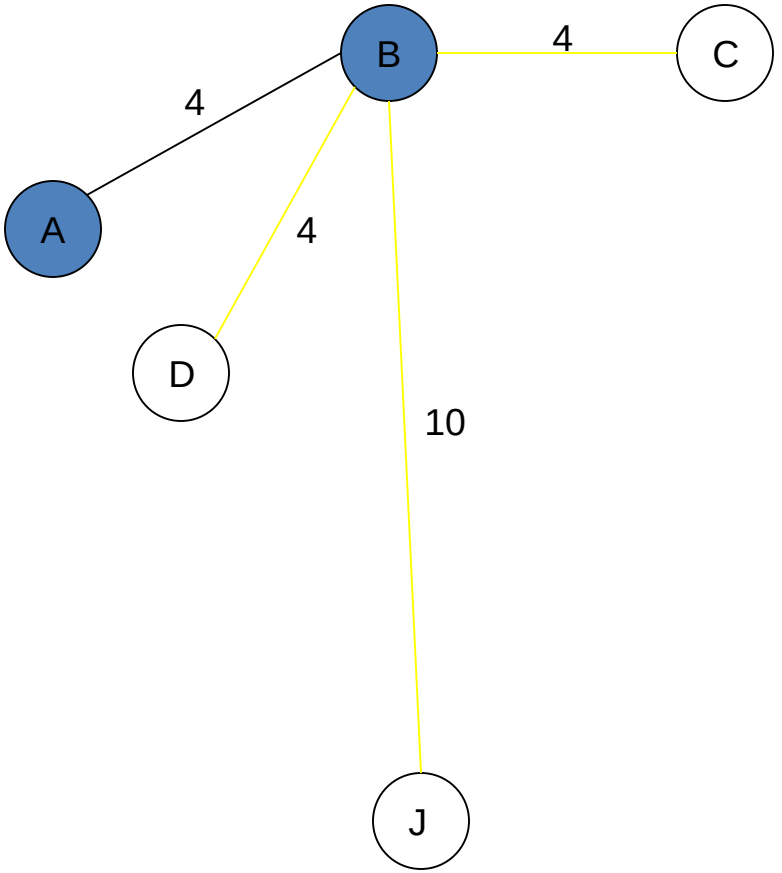


Αλγόριθμος Boruvka – Παράδειγμα VI

1ος γύρος

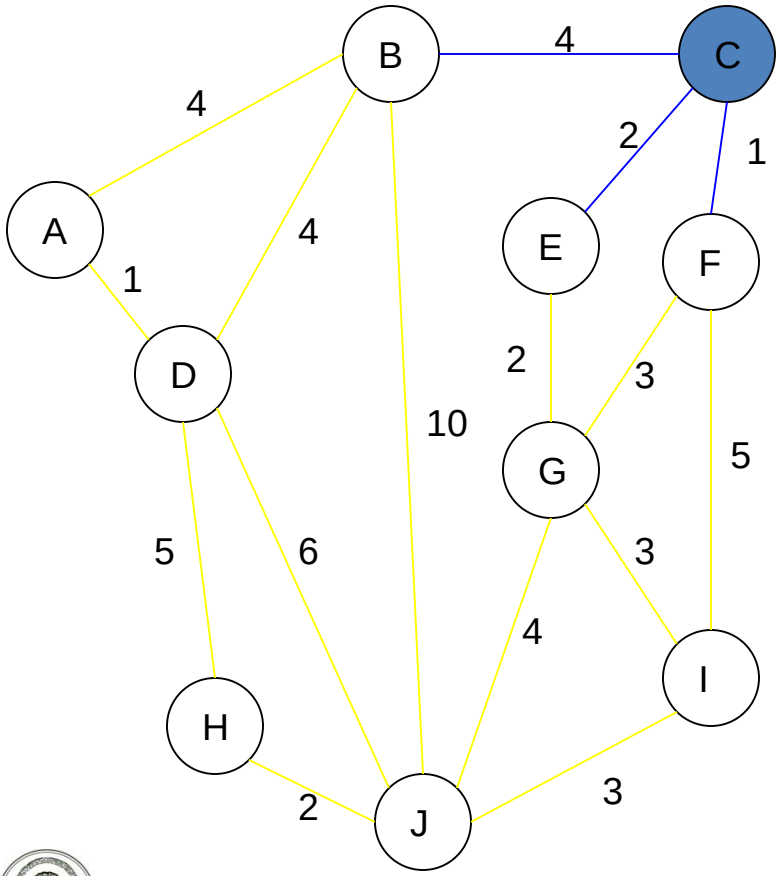


Ακμή B-A

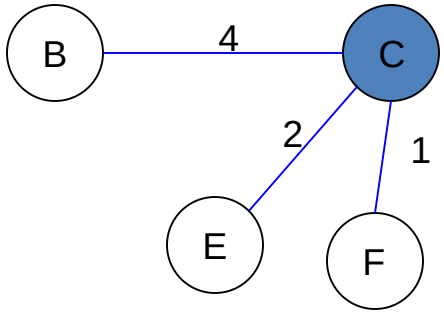


Αλγόριθμος Boruvka – Παράδειγμα VII

1ος γύρος

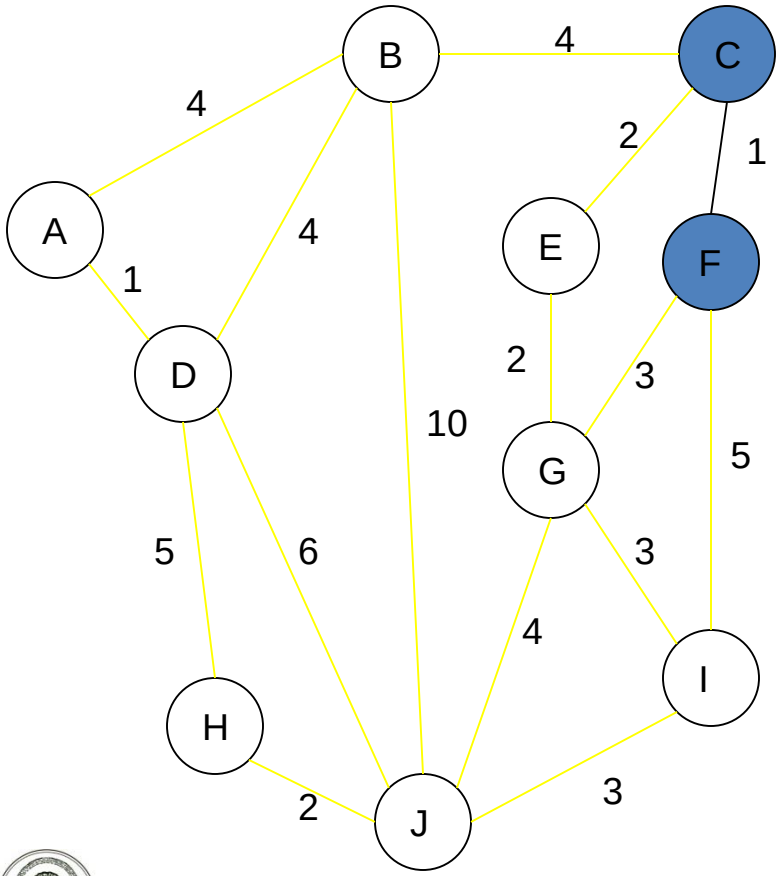


Δένδρο C

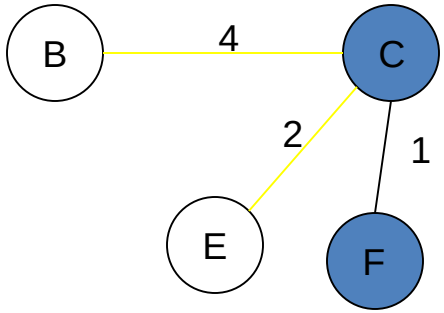


Αλγόριθμος Boruvka – Παράδειγμα VIII

1ος γύρος

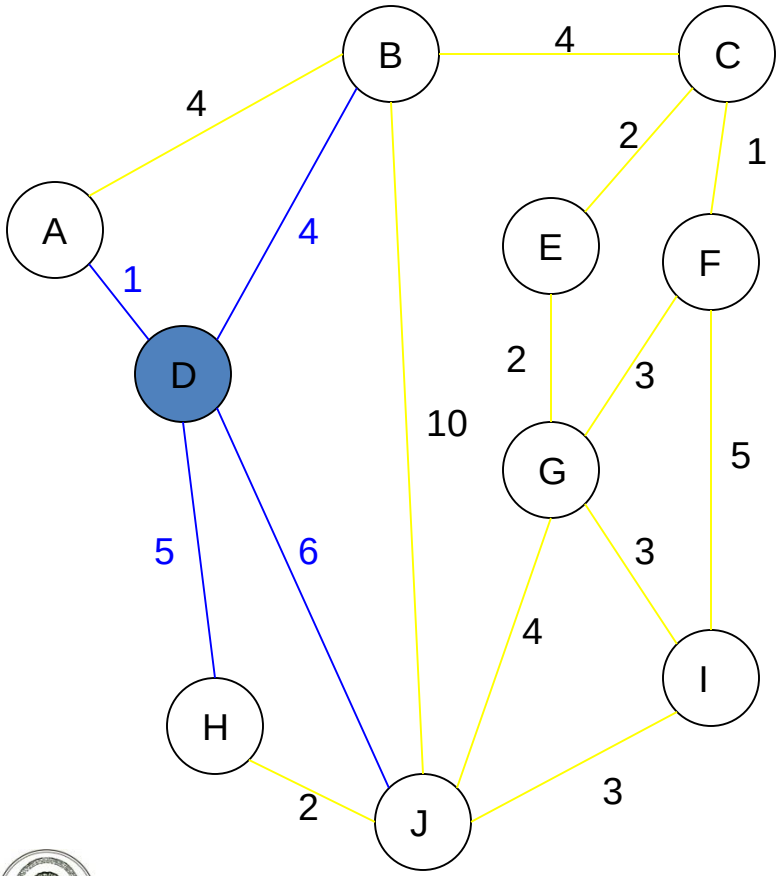


Ακμή C-F

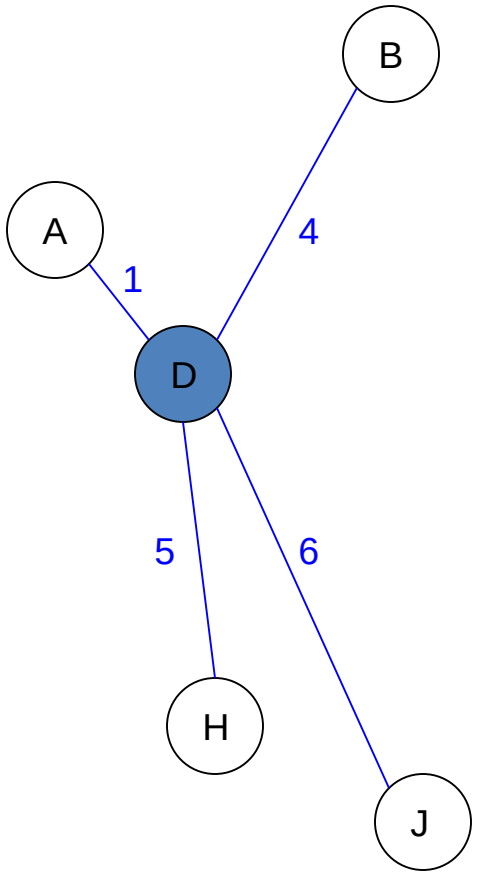


Αλγόριθμος Boruvka – Παράδειγμα IX

1ος γύρος

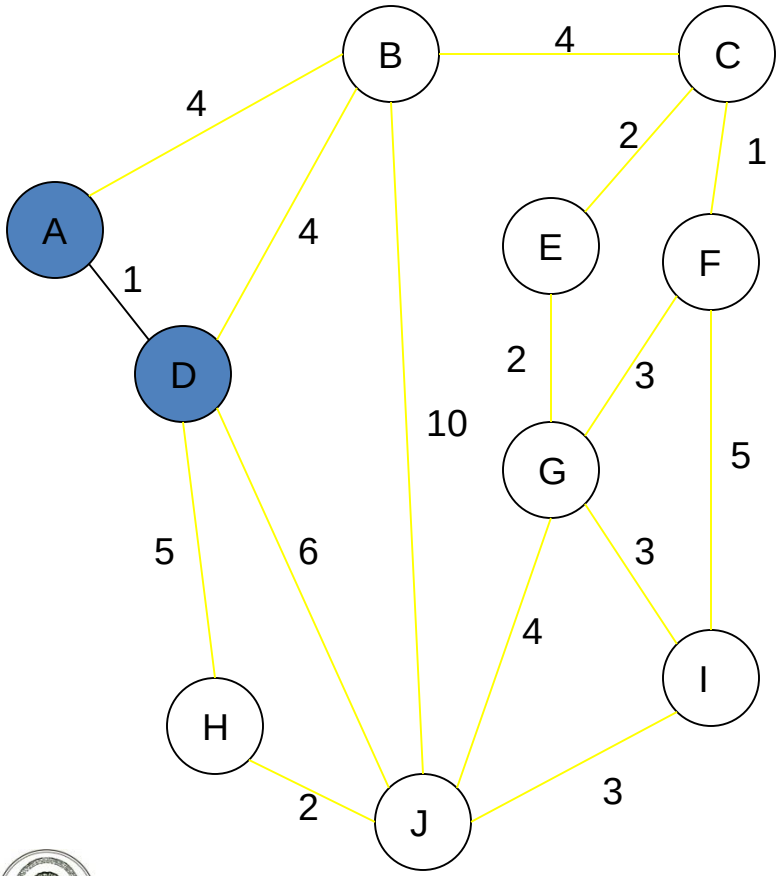


Δένδρο D

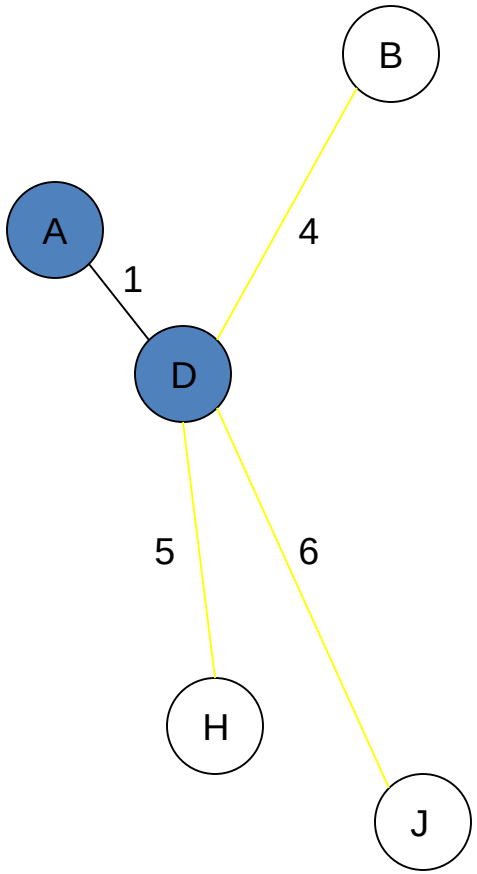


Αλγόριθμος Boruvka – Παράδειγμα X

1ος γύρος

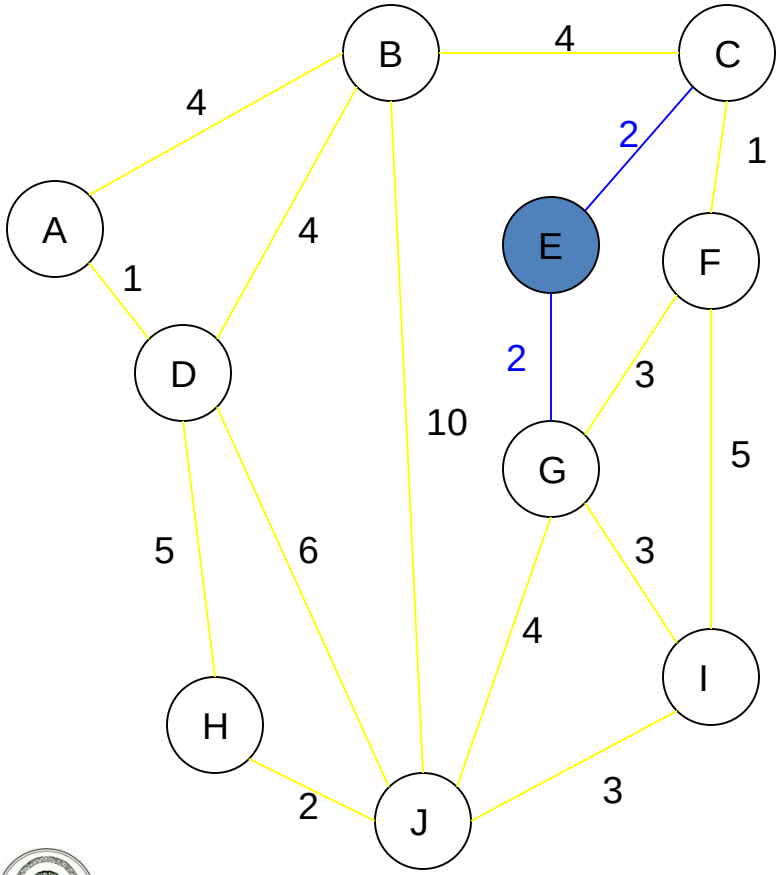


Ακμή D-A

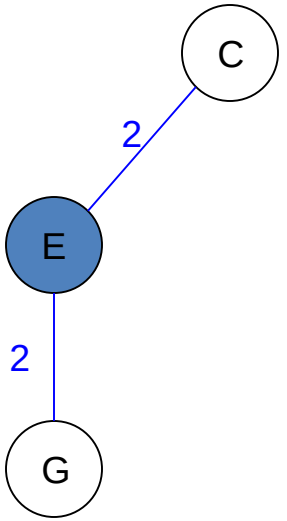


Αλγόριθμος Boruvka – Παράδειγμα XI

1ος γύρος

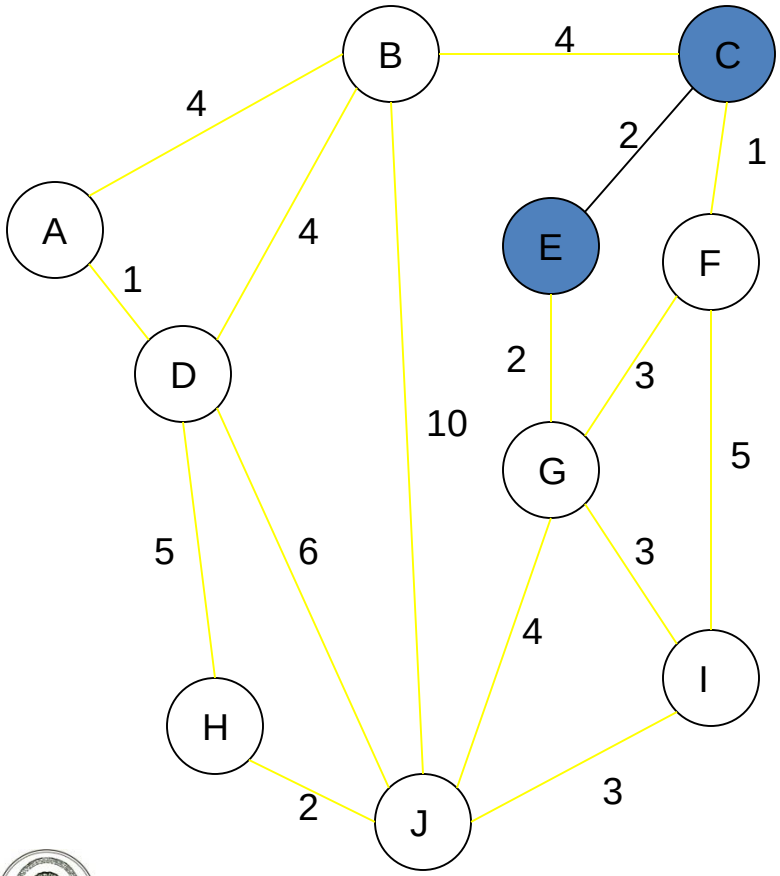


Δένδρο E

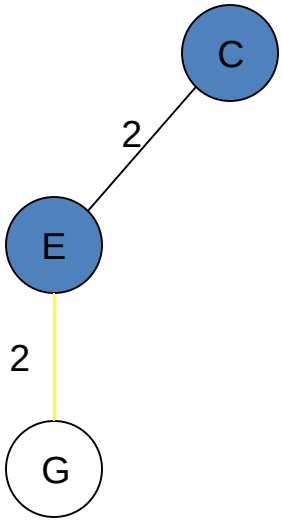


Αλγόριθμος Boruvka – Παράδειγμα XII

1ος γύρος

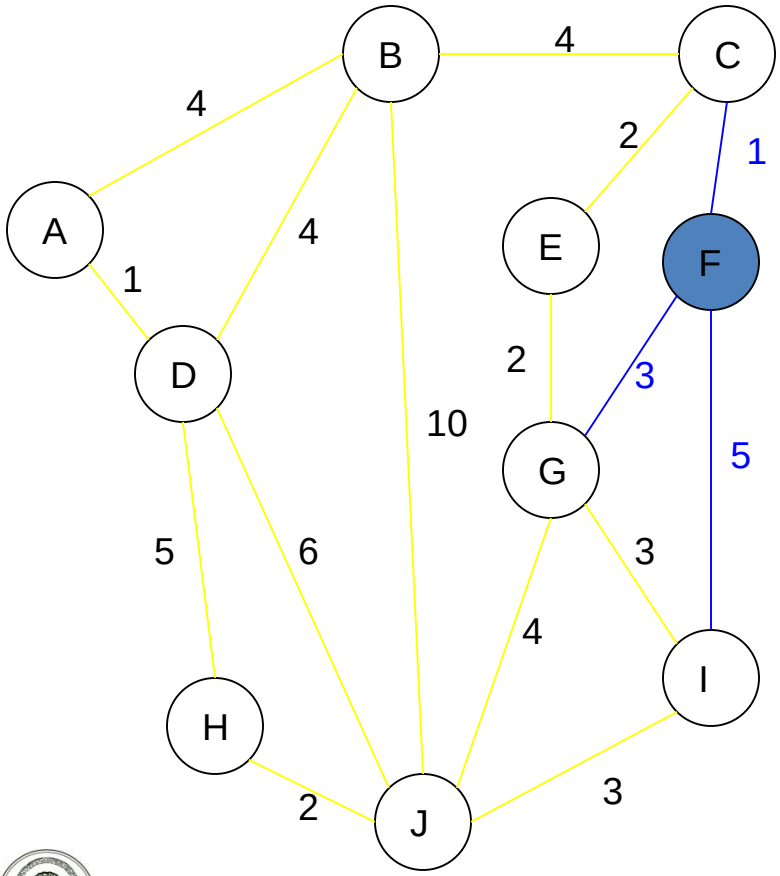


Ακμή E-C

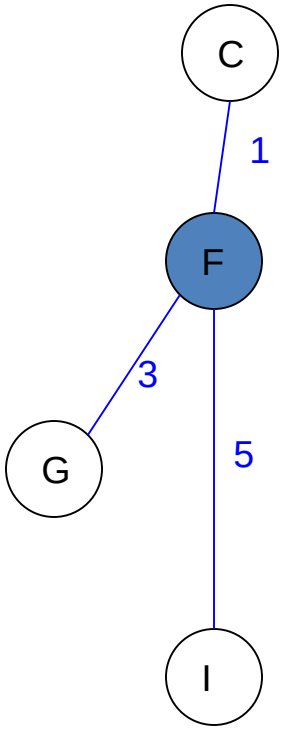


Αλγόριθμος Boruvka – Παράδειγμα XIII

1ος γύρος

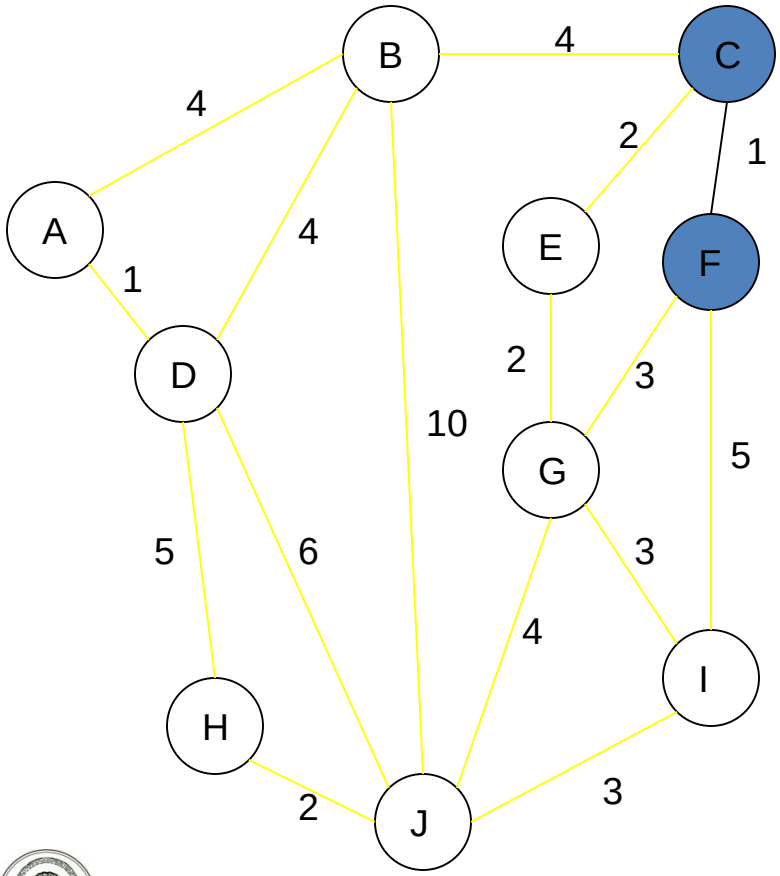


Δένδρο F

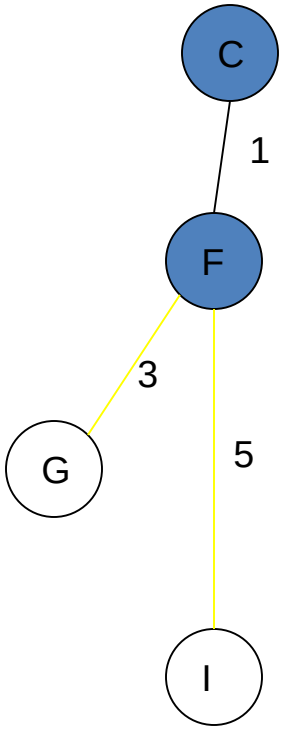


Αλγόριθμος Boruvka – Παράδειγμα XIV

1ος γύρος

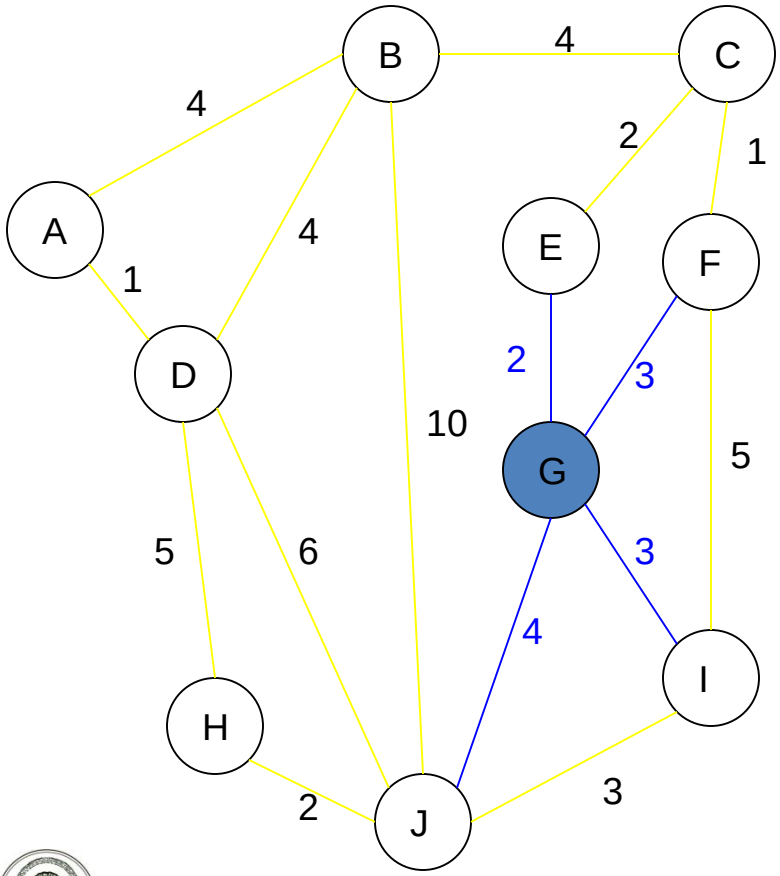


Ακμή F-C

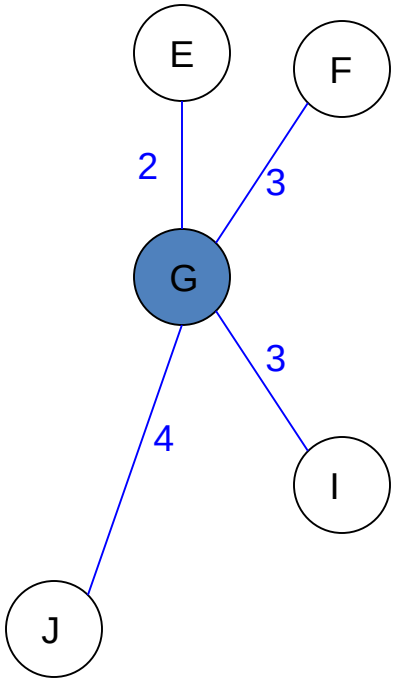


Αλγόριθμος Boruvka – Παράδειγμα XV

1ος γύρος

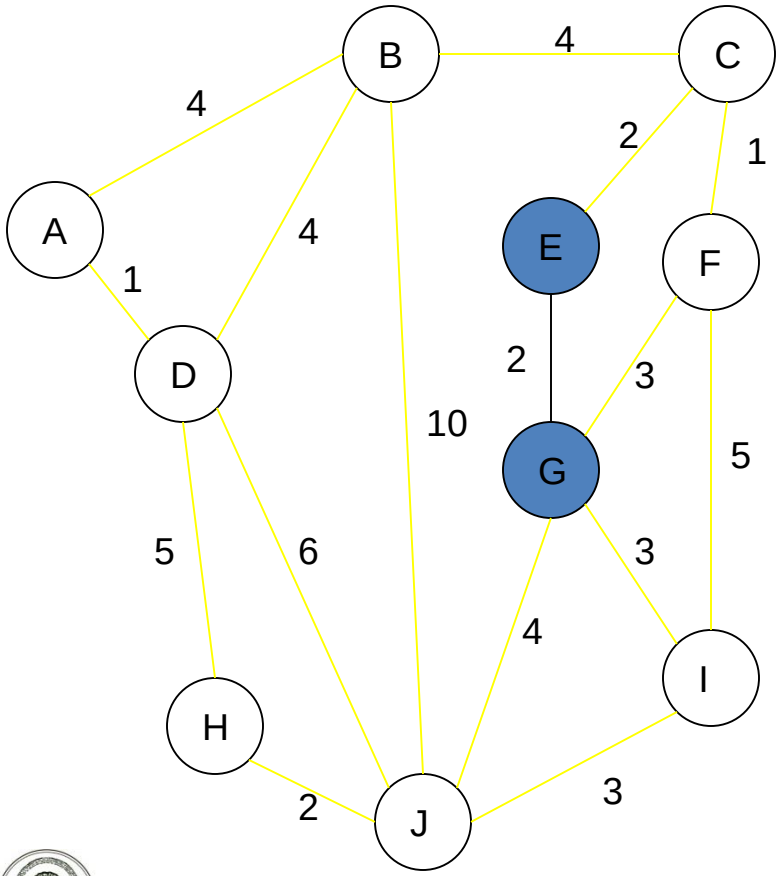


Δένδρο G

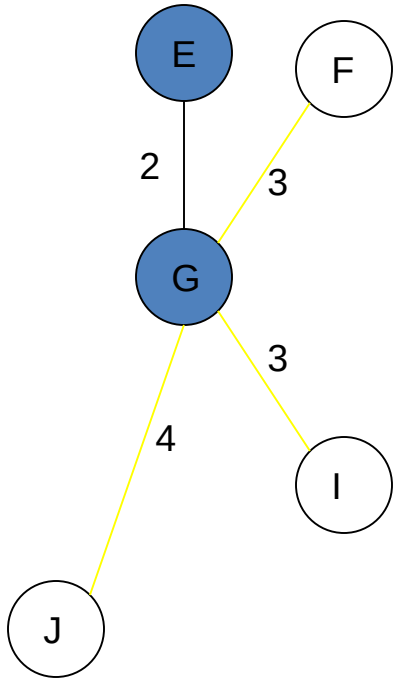


Αλγόριθμος Boruvka – Παράδειγμα XVI

1ος γύρος

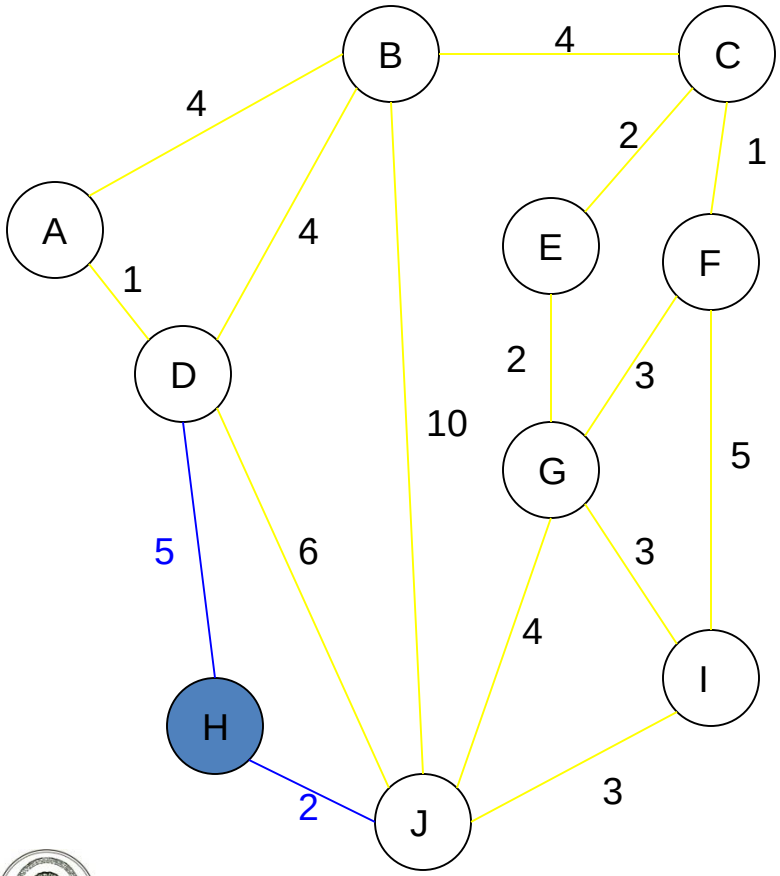


Ακμή G-E

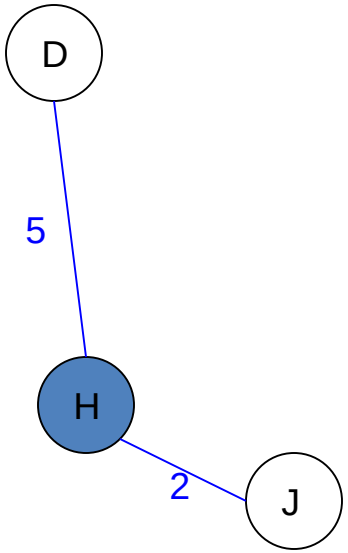


Αλγόριθμος Boruvka – Παράδειγμα XVII

1ος γύρος

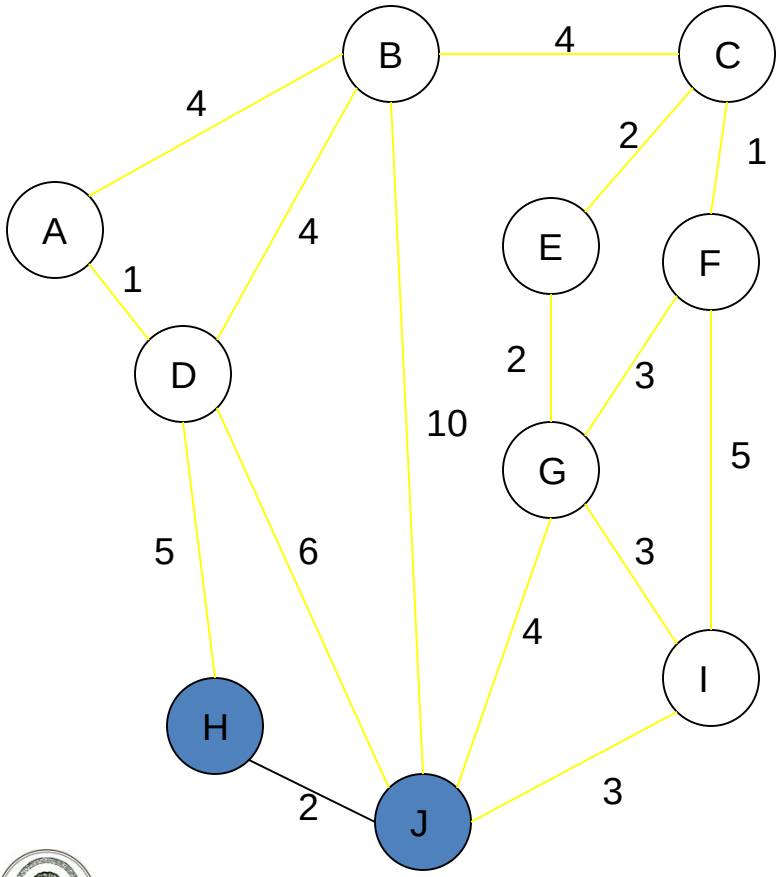


Δένδρο Η

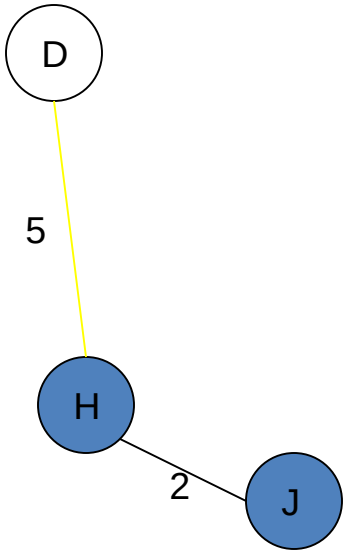


Αλγόριθμος Boruvka – Παράδειγμα XVIII

1ος γύρος

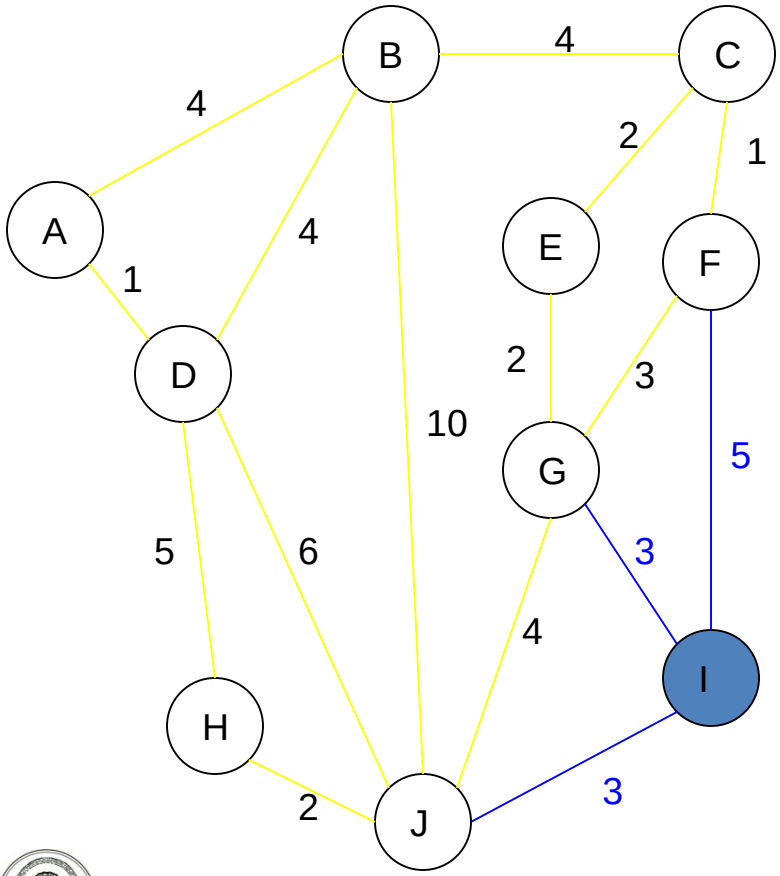


Ακμή H-J

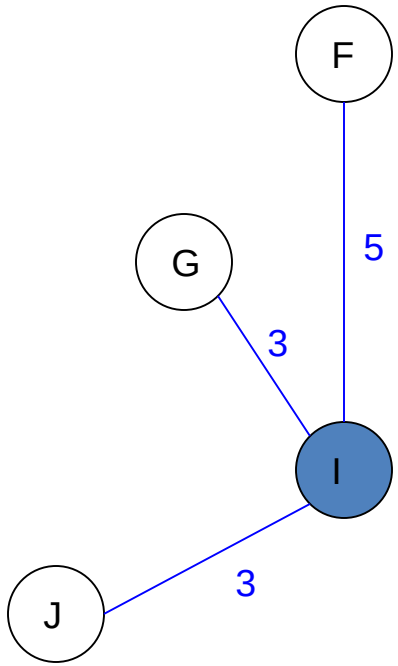


Αλγόριθμος Boruvka – Παράδειγμα XIX

1ος γύρος

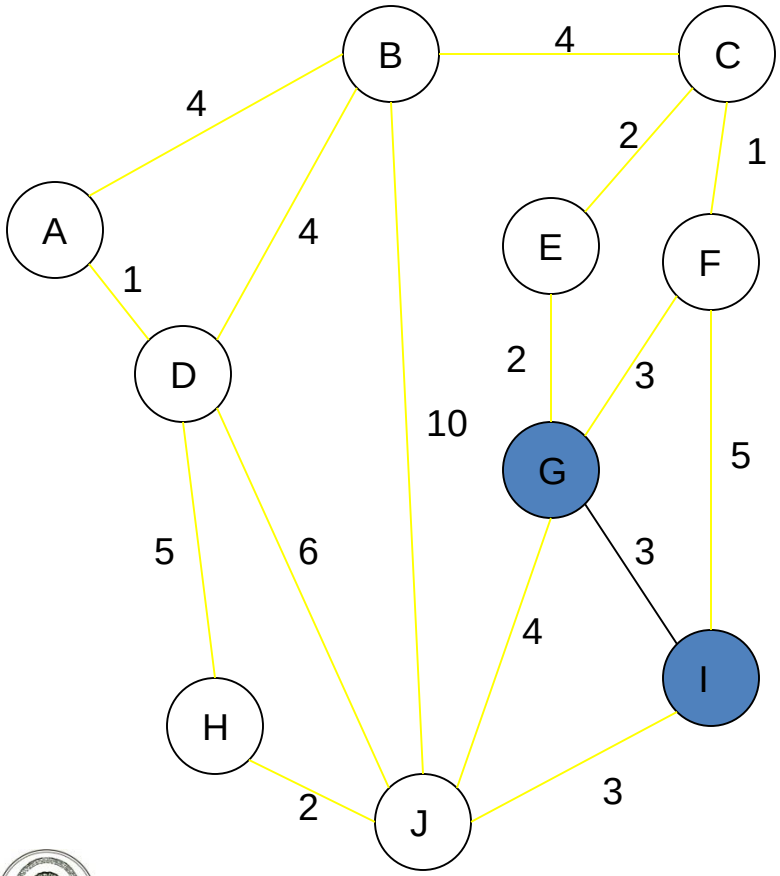


Δένδρο I

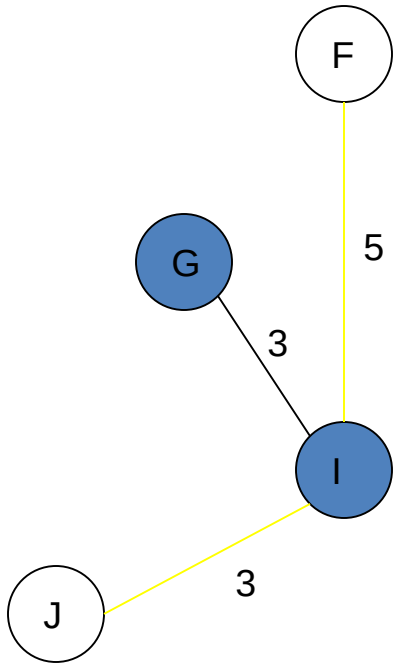


Αλγόριθμος Boruvka – Παράδειγμα XX

1ος γύρος

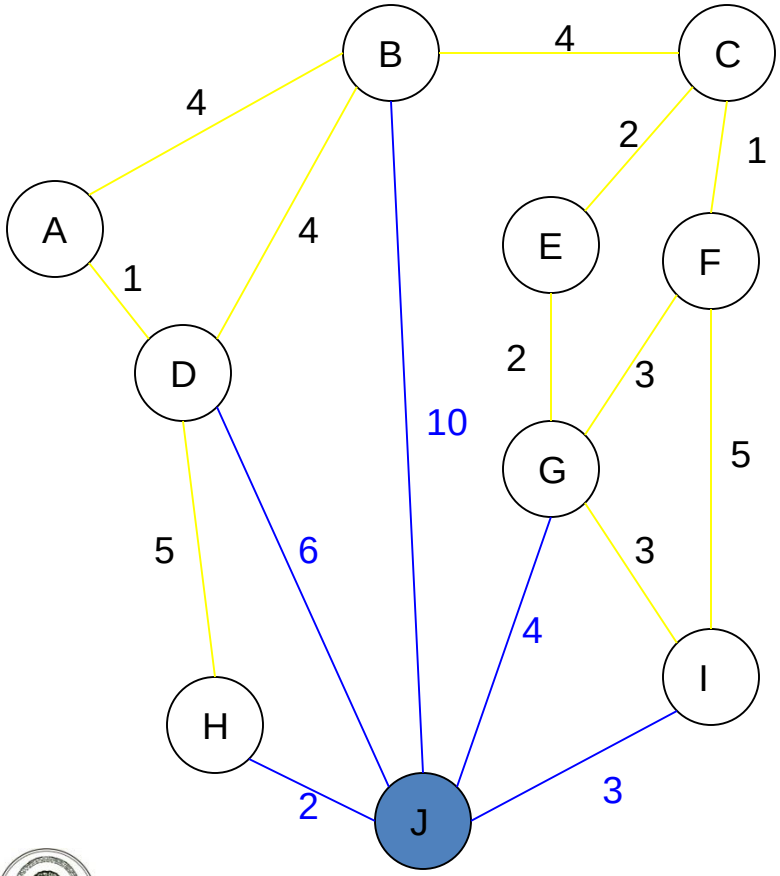


Ακμή I-G

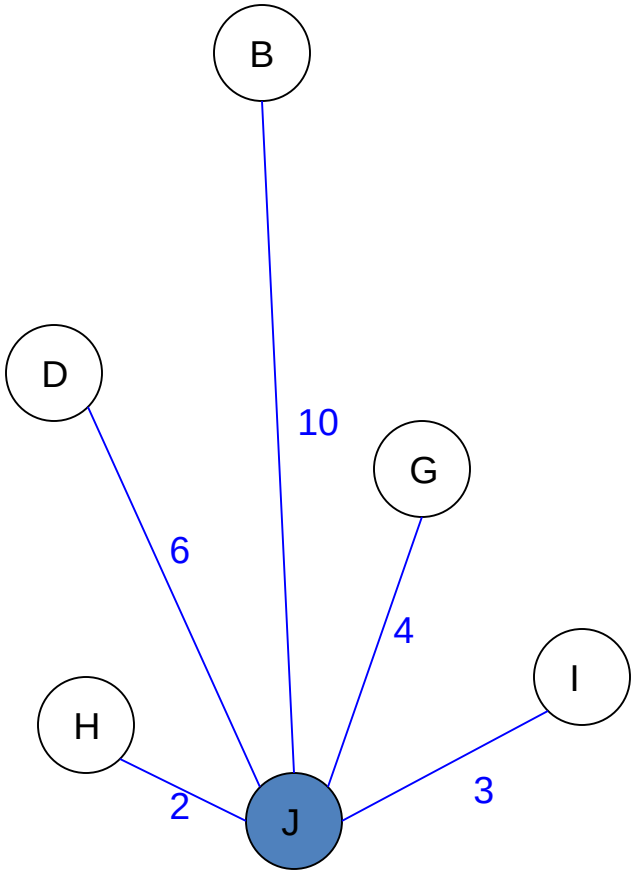


Αλγόριθμος Boruvka – Παράδειγμα XXI

1ος γύρος

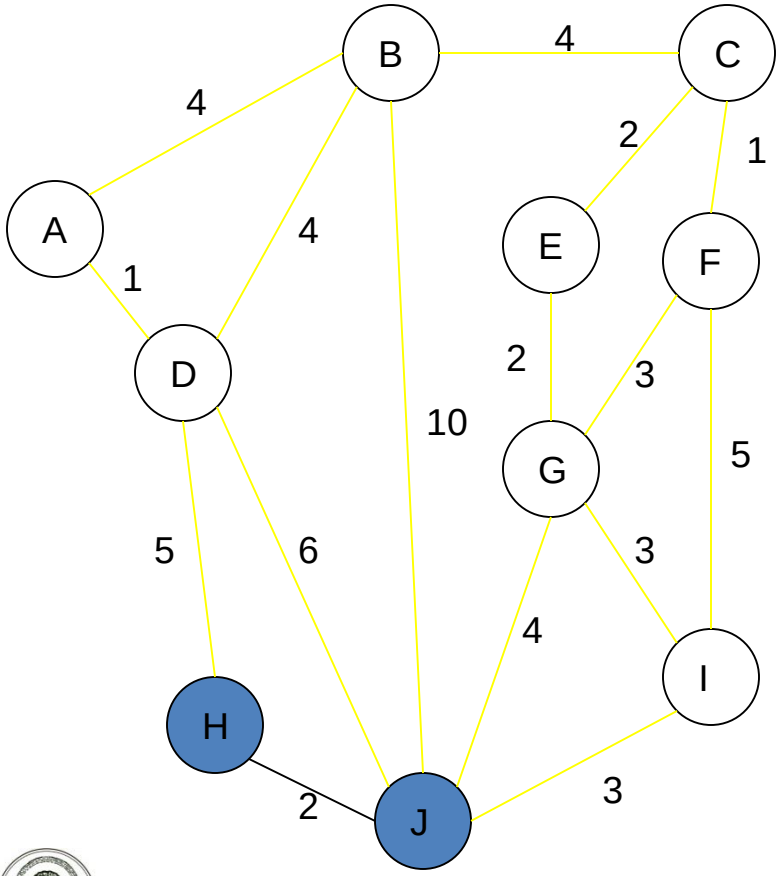


Δένδρο J

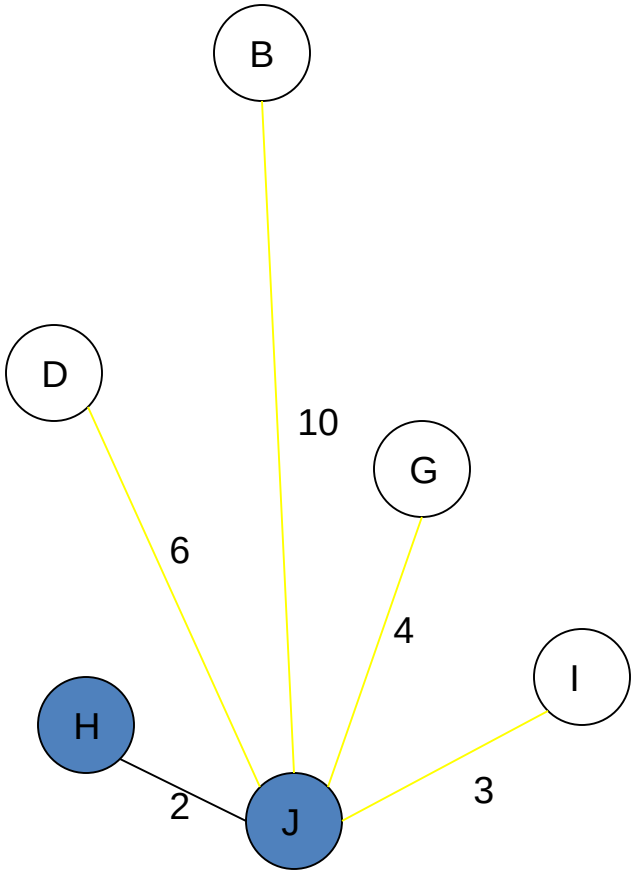


Αλγόριθμος Boruvka – Παράδειγμα XXII

1ος γύρος

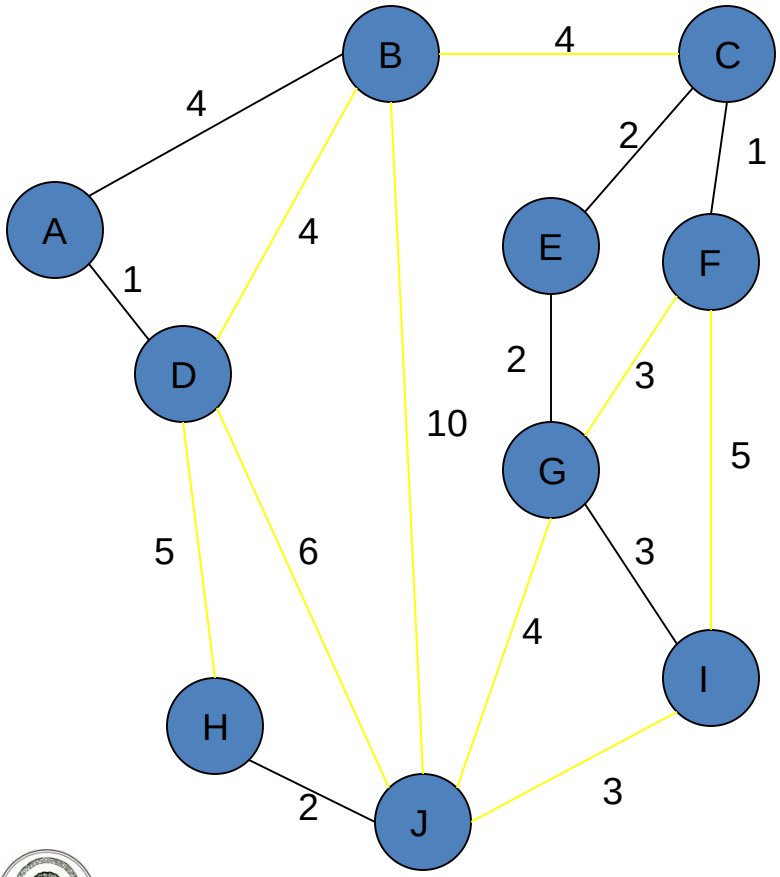


Ακμή J-H



Αλγόριθμος Boruvka – Παράδειγμα XXIII

Τέλος 1ου γύρου – προσθήκη ακμών

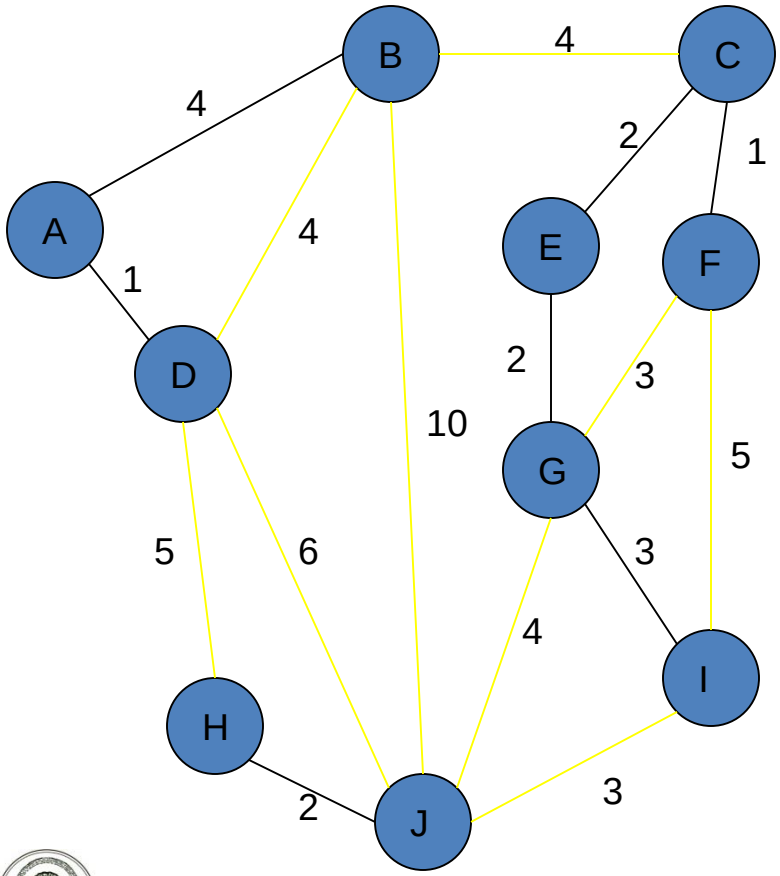


Λίστα ακμών προς προσθήκη

- A-D
- B-A
- C-F
- D-A
- E-C
- F-C
- G-E
- H-J
- I-G
- J-H

Αλγόριθμος Boruvka – Παράδειγμα XXIV

Δένδρα στην αρχή του 2ου γύρου

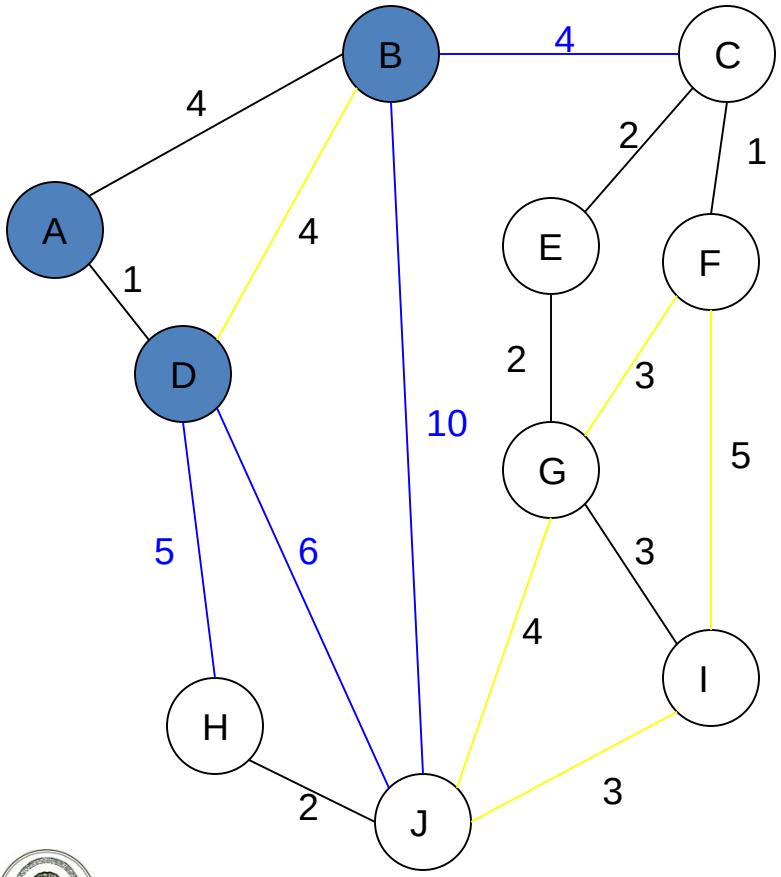


Λίστα δένδρων

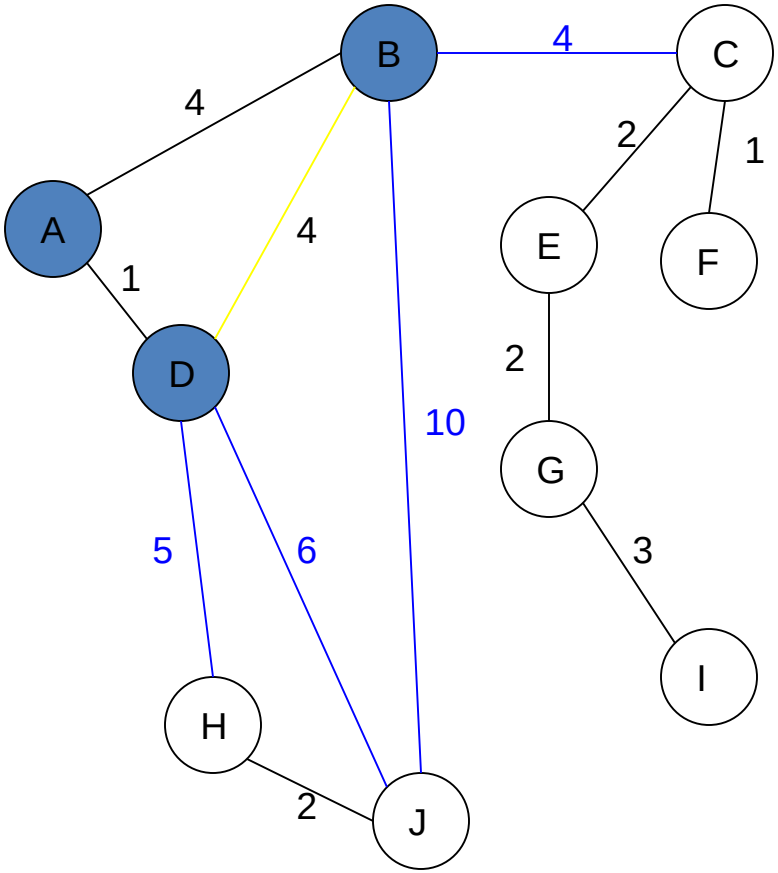
- D-A-B
- F-C-E-G-I
- H-J

Αλγόριθμος Boruvka – Παράδειγμα XXV

2ος γύρος

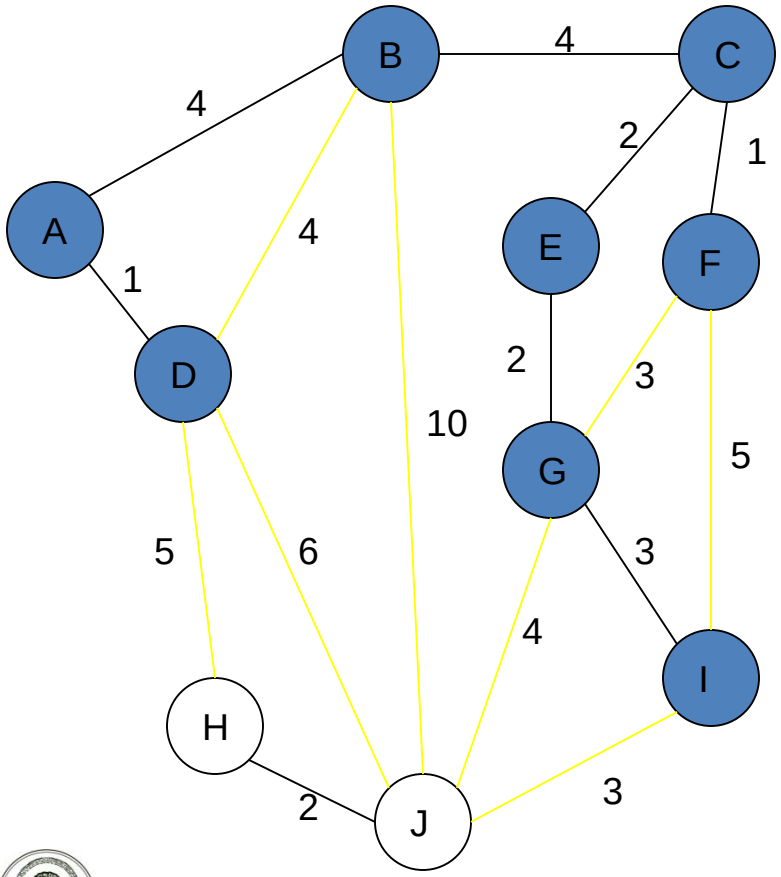


Δένδρο D-A-B

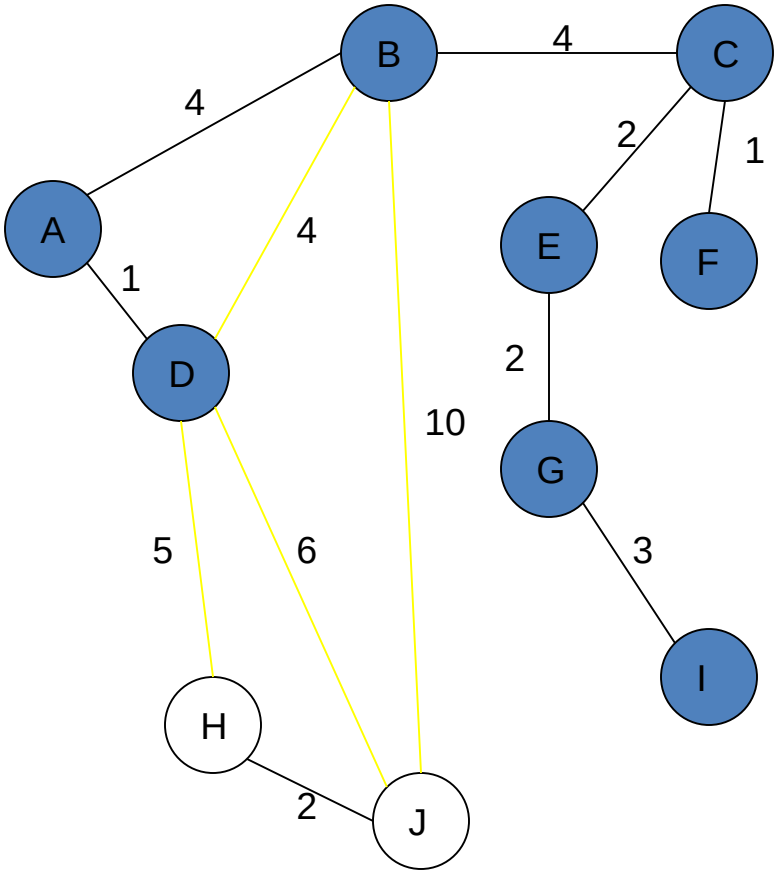


Αλγόριθμος Boruvka – Παράδειγμα XXVI

2ος γύρος

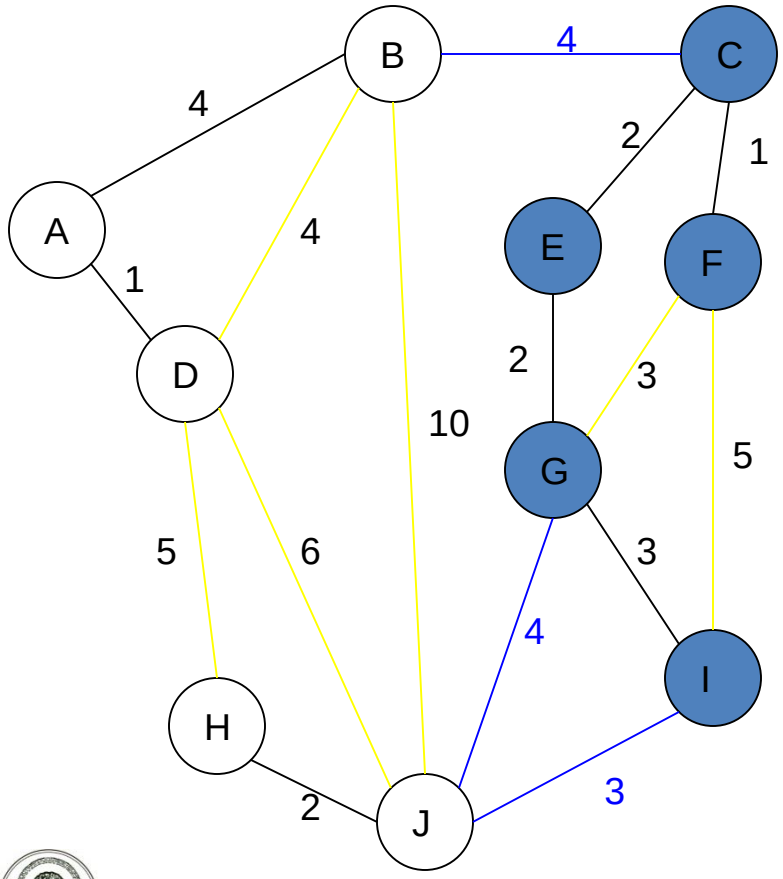


Ακμή B-C

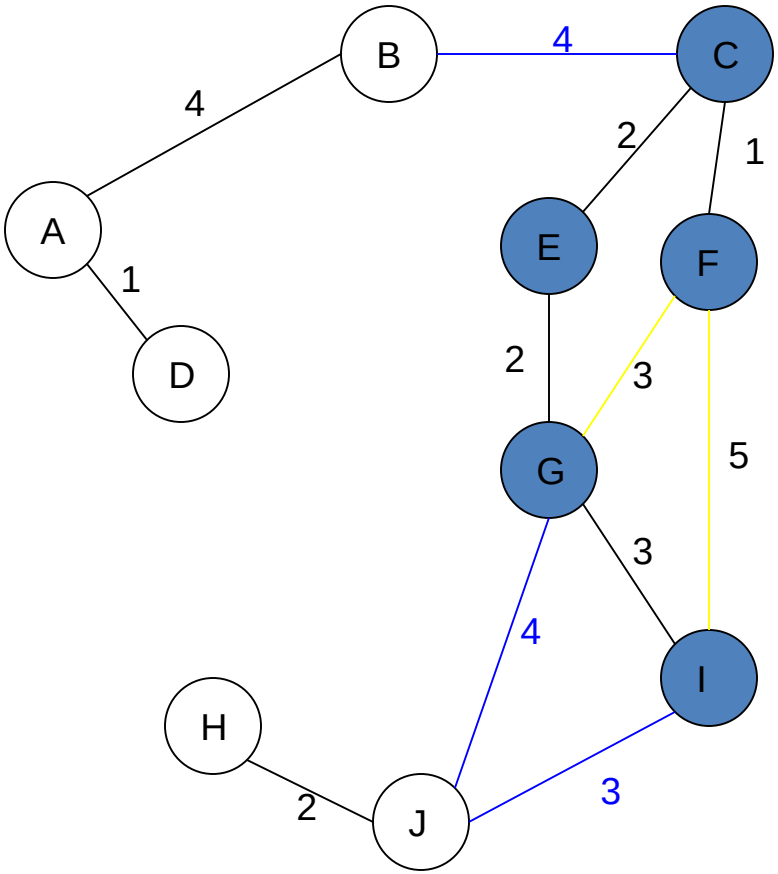


Αλγόριθμος Boruvka – Παράδειγμα XXVII

2ος γύρος

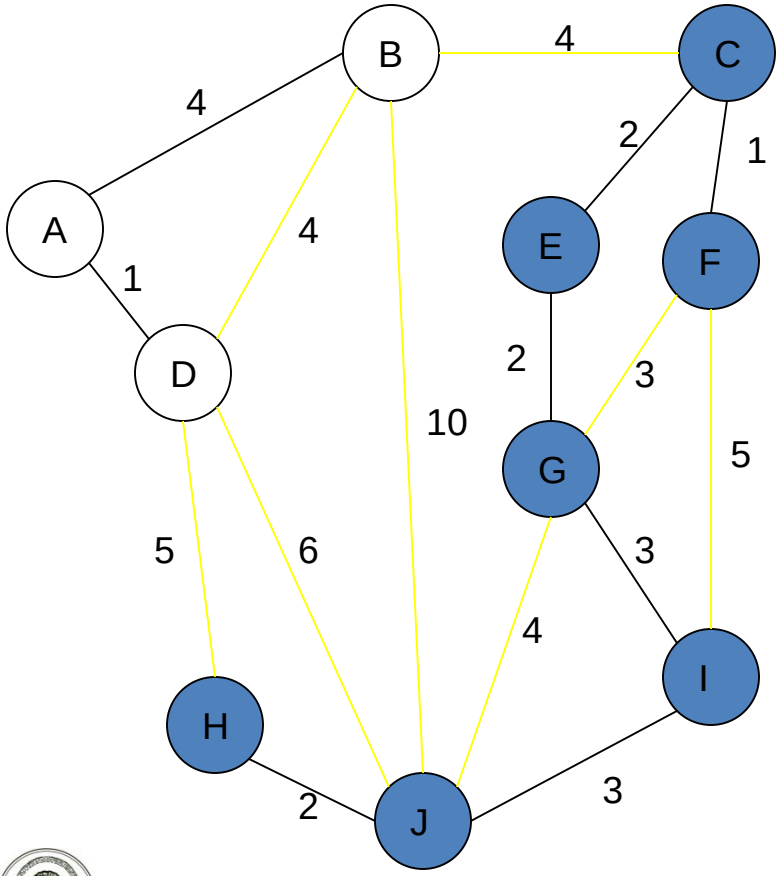


Δένδρο F-C-E-G-I

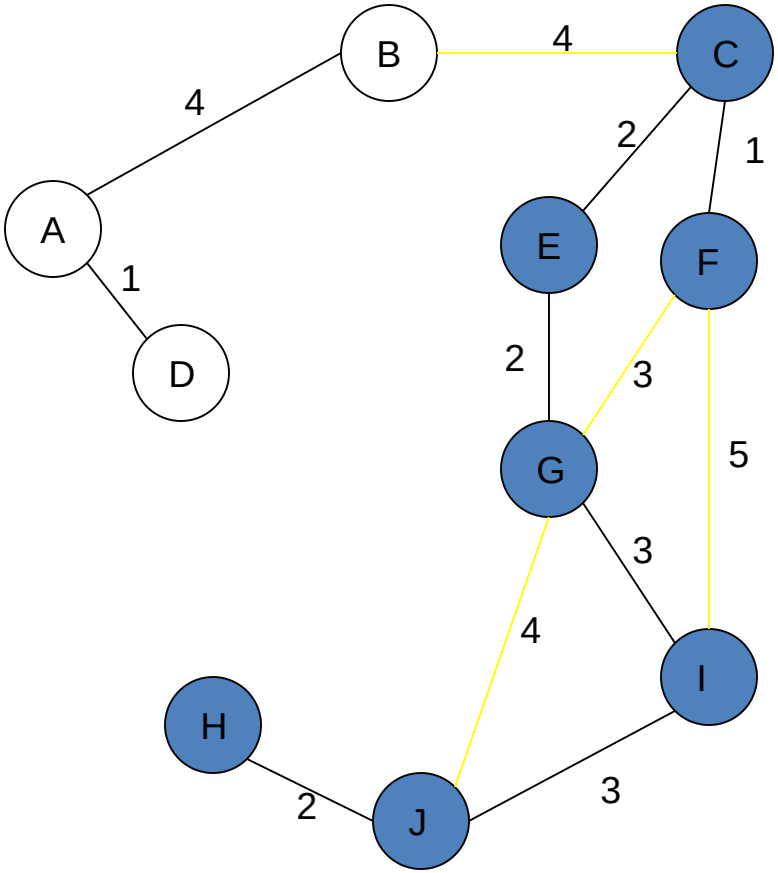


Αλγόριθμος Boruvka – Παράδειγμα XXVIII

2ος γύρος

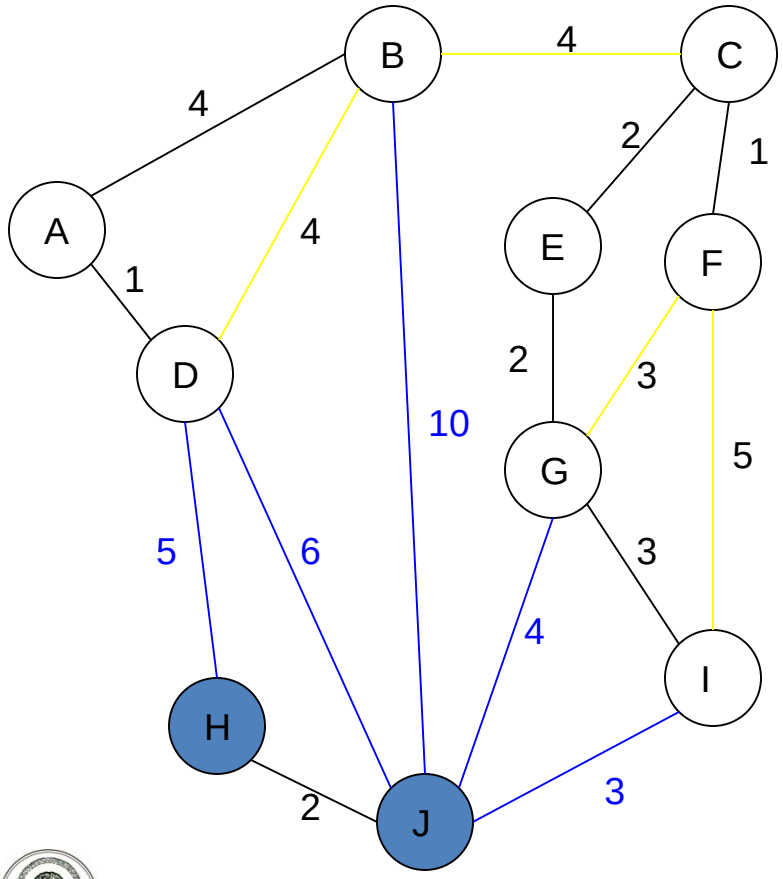


Ακμή I-J

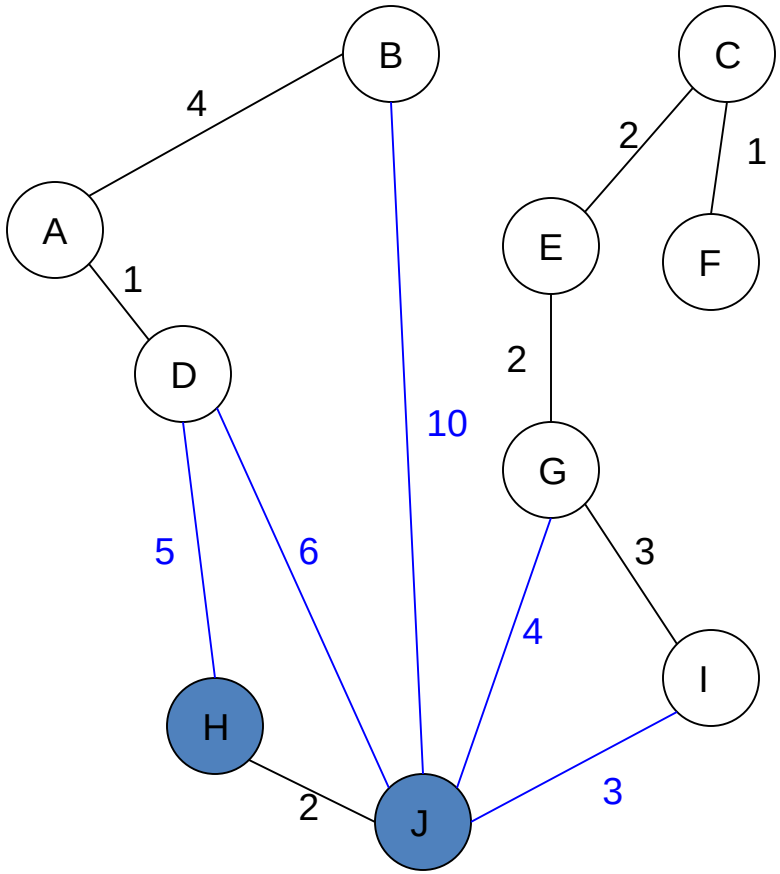


Αλγόριθμος Boruvka – Παράδειγμα XXIX

2ος γύρος

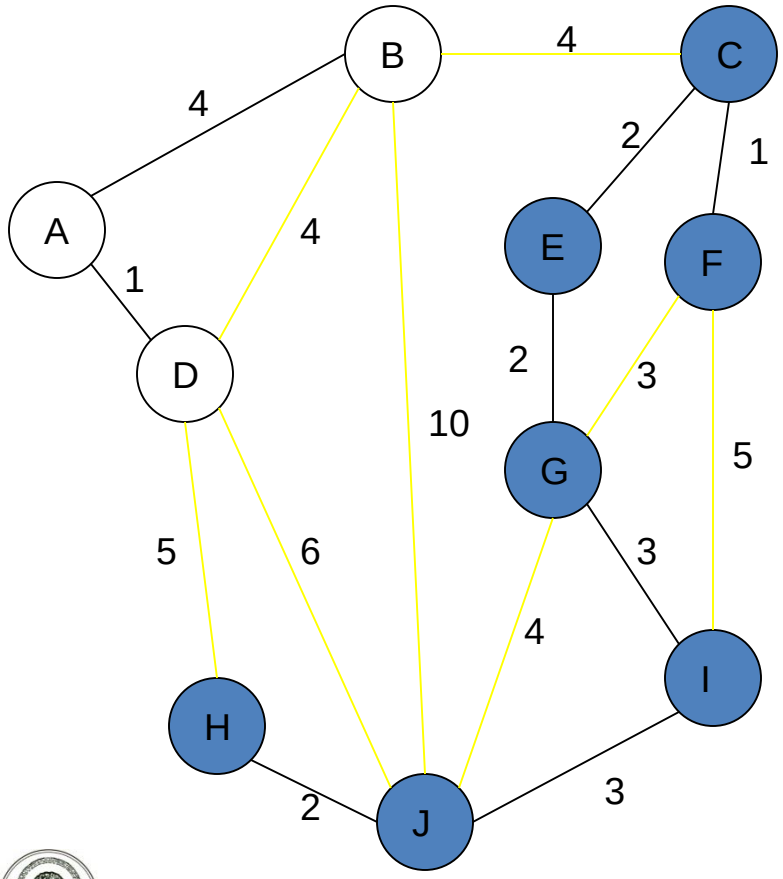


Δένδρο H-J

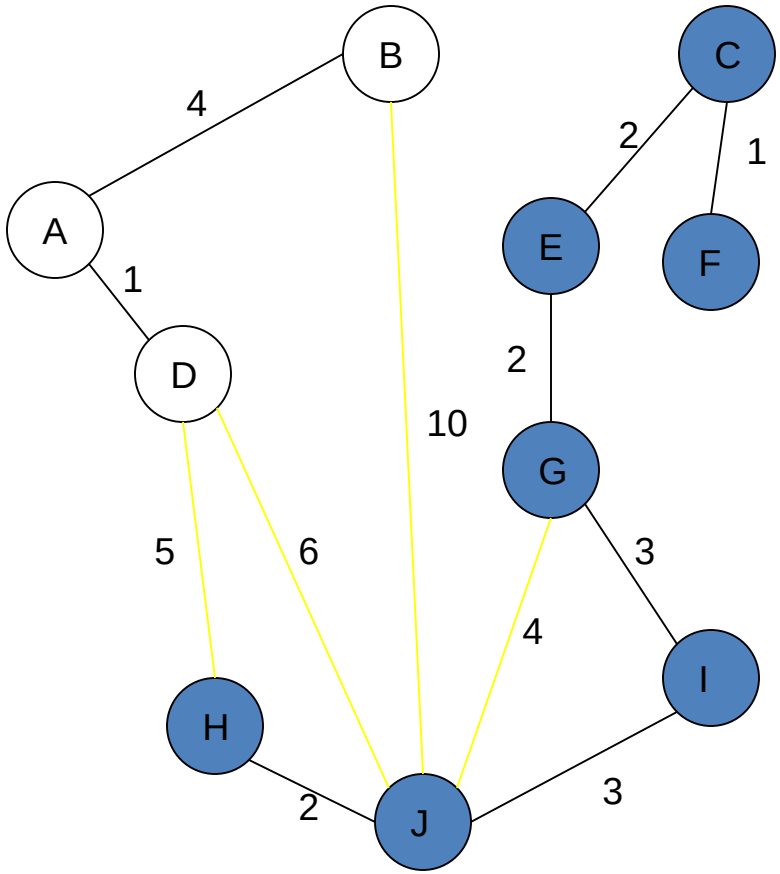


Αλγόριθμος Boruvka – Παράδειγμα XXX

2ος γύρος

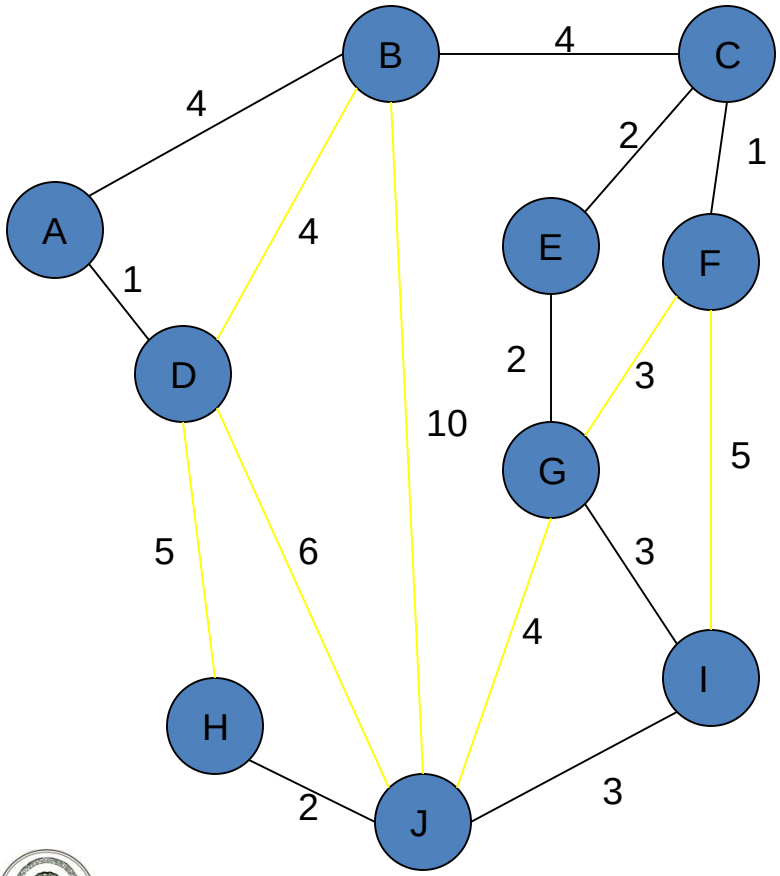


Ακμή J-I



Αλγόριθμος Boruvka – Παράδειγμα XXXI

Τέλος 2ου γύρου
Προσθήκη ακμών

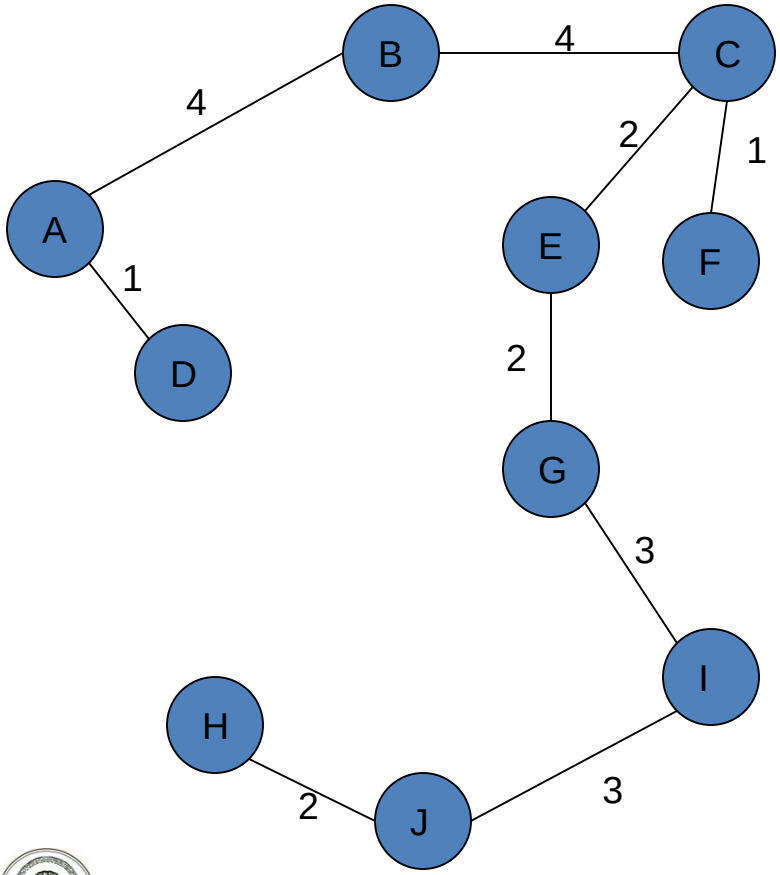


Λίστα ακμών
προς προσθήκη

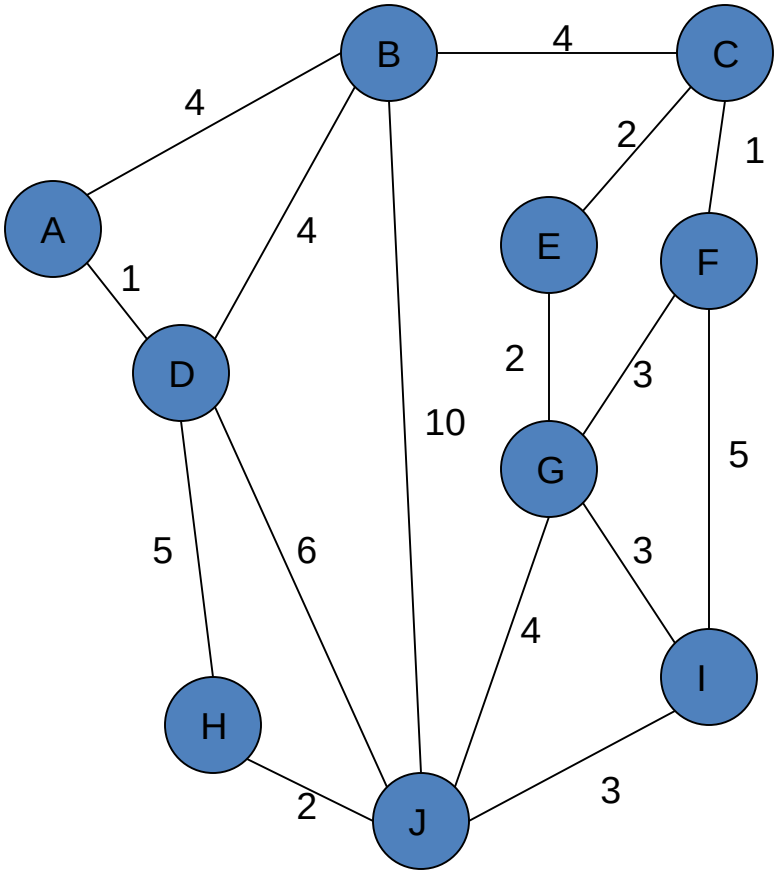
- B-C
- I-J
- J-I

Αλγόριθμος Boruvka – Παράδειγμα XXXII

Ελάχιστο ζευγνύον δένδρο



Πλήρης γράφος



Ανάλυση του Αλγορίθμου Boruvka

- Χρόνος Εκτέλεσης = O(m log n) (m=ακμές, n=κόμβοι)
- Αν και αυτός ο αλγόριθμος είναι δύσκολος να επεξηγηθεί, αντίθετα με τους δύο προηγούμενους αλγόριθμους, δεν χρειάζεται κάποια πολύπλοκη δομή δεδομένων.
- Όπως ο αλγόριθμος του Prim, δεν ασχολείται με τον έλεγχο κύκλων. Όμως χρειάζεται να «δει» ολόκληρο το γράφο, ελέγχοντας κομμάτια του.
- Όπως ο αλγόριθμος του Kruskal είναι καλύτερο αν οι ακμές διατηρούνται στο ελάχιστο (αν και δεν πειράζει να διατηρήσουμε τους κόμβους στο ελάχιστο επίσης).

Συμπεράσματα

- Οι αλγόριθμοι των Kruskal και Boruvka έχουν καλύτερους χρόνους εκτέλεσης αν το πλήθος των ακμών διατηρείται χαμηλά, ενώ ο αλγόριθμος του Prim έχει καλύτερο χρόνο εκτέλεσης εάν και το πλήθος των ακμών και το πλήθος των κόμβων είναι μικρό.
- Ο αλγόριθμος του Boruvka αποφεύγει τις πολύπλοκες δομές δεδομένων που απαιτούνται από τους άλλους δύο αλγορίθμους.
- Επομένως ο καλύτερος αλγόριθμος εξαρτάται από το γράφο δεδομένου του κόστους των πολύπλοκων δεδομένων.
- Ο καλύτερος αλγόριθμος είναι ένα υβριδικός των Boruvka και Prim, ο οποίος δεν εξετάζεται εδώ. Κάνει O(log log n) περάσματα του Boruvka αλγορίθμου και στη συνέχεια αλλάζει στον αλγόριθμο του Prim, με αποτέλεσμα O(m log log n) χρόνο εκτέλεσης. Επομένως είναι και ο γρηγορότερος αλγόριθμος, αλλά θα χρειαστεί Fibonacci σωρό για τον αλγόριθμο του Prim, το οποίο αποφεύγει ο αλγόριθμος του Boruvka όταν χρησιμοποιείται μόνος του.

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Παρόμοια Διανομή [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] http://creativecommons.org/licenses/by-sa/4.0/

Τέλος ενότητας

Επεξεργασία: Ανδρέας Κοσματόπουλος

Θεσσαλονίκη, Μάρτιος 2015

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.