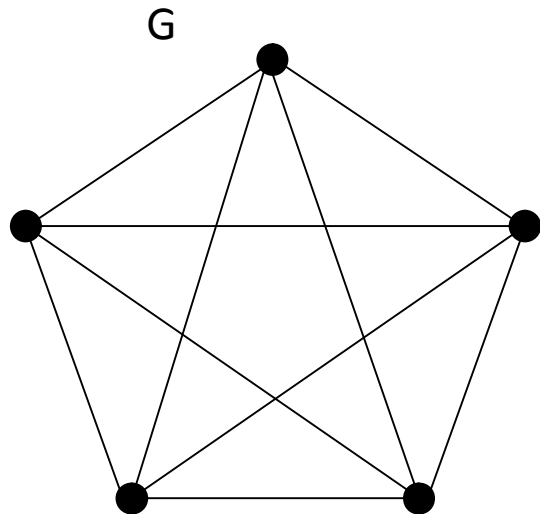


Αλγόριθμοι εύρεσης ελάχιστων γεννητικών δέντρων (MST)

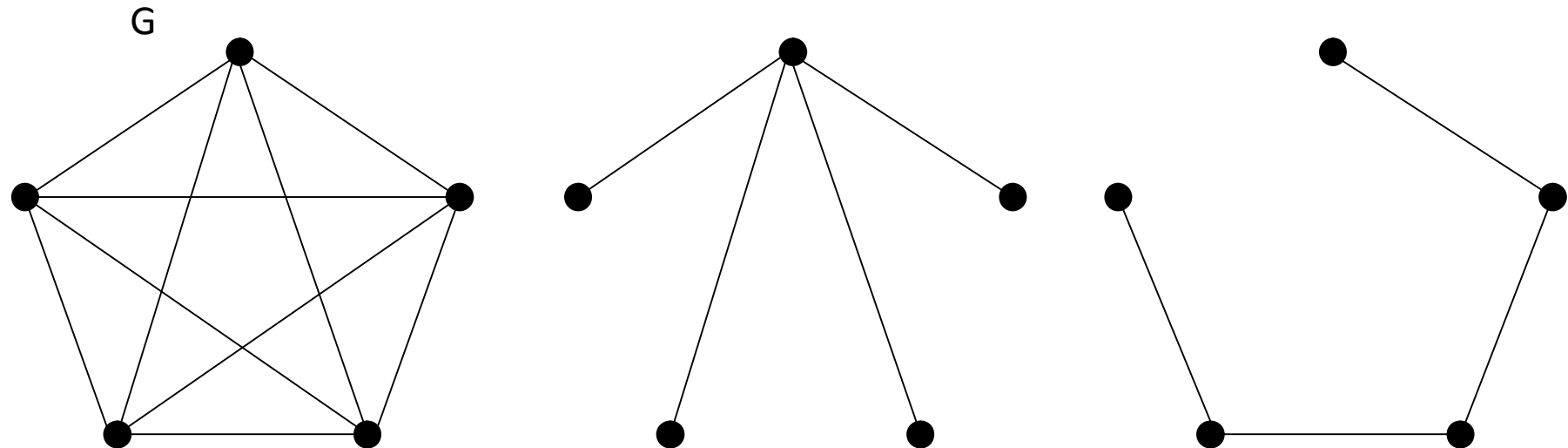
Γεννητικό δέντρο (Spanning Tree)

- Ένα γεννητικό δέντρο για ένα γράφημα G είναι
 - ένα υπογράφημα του G που είναι δέντρο (δηλ., είναι συνεκτικό και δεν περιέχει κύκλους) και
 - περιέχει όλες τις κορυφές του G
- Ένα γεννητικό δέντρο για κάθε γράφημα G με n κορυφές έχει $n-1$ ακμές



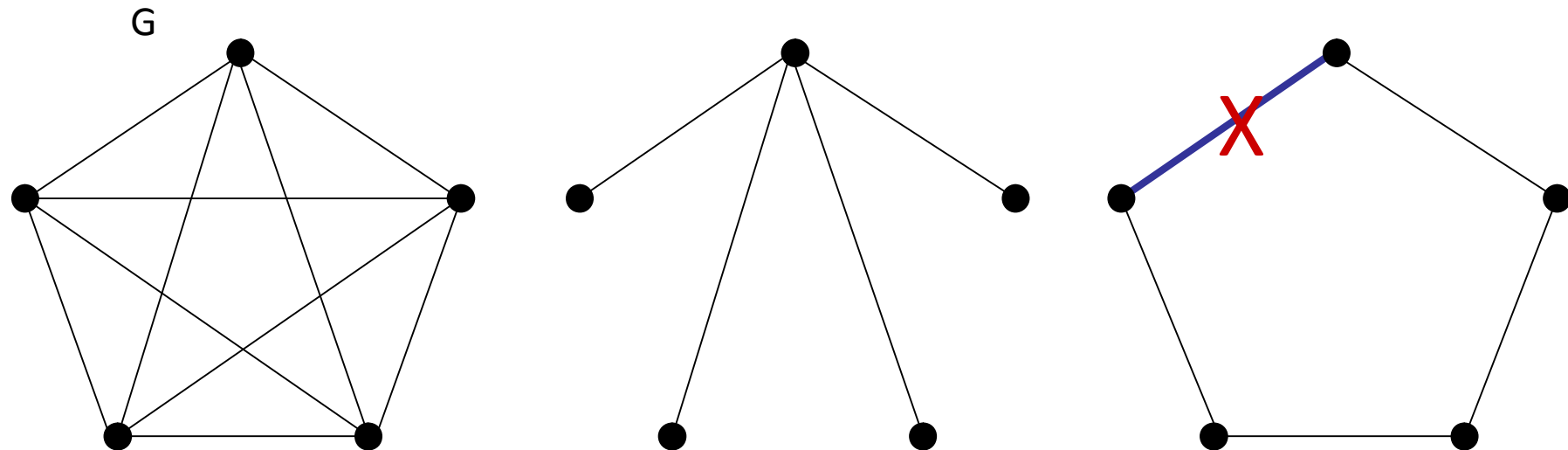
Γεννητικό δέντρο (Spanning Tree)

- Ένα γεννητικό δέντρο για ένα γράφημα G είναι
 - ένα υπογράφημα του G που είναι δέντρο (δηλ., είναι συνεκτικό και δεν περιέχει κύκλους) και
 - περιέχει όλες τις κορυφές του G
- Ένα γεννητικό δέντρο για κάθε γράφημα G με n κορυφές έχει $n-1$ ακμές



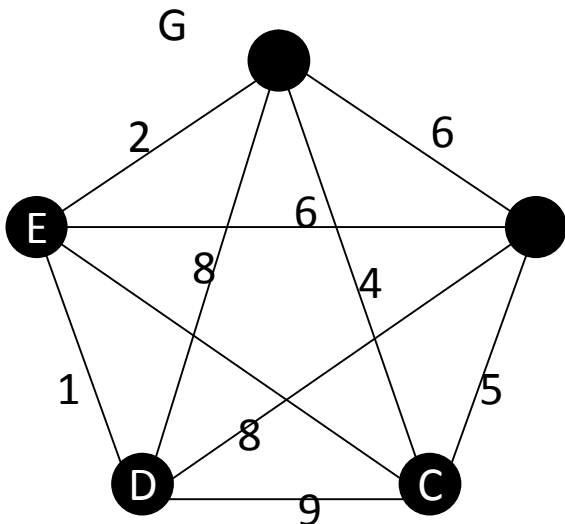
Γεννητικό δέντρο (Spanning Tree)

- Ένα γεννητικό δέντρο για ένα γράφημα G είναι
 - ένα υπογράφημα του G που είναι δέντρο (δηλ., είναι συνεκτικό και δεν περιέχει κύκλους) και
 - περιέχει όλες τις κορυφές του G
- Ένα γεννητικό δέντρο για κάθε γράφημα G με n κορυφές έχει $n-1$ ακμές



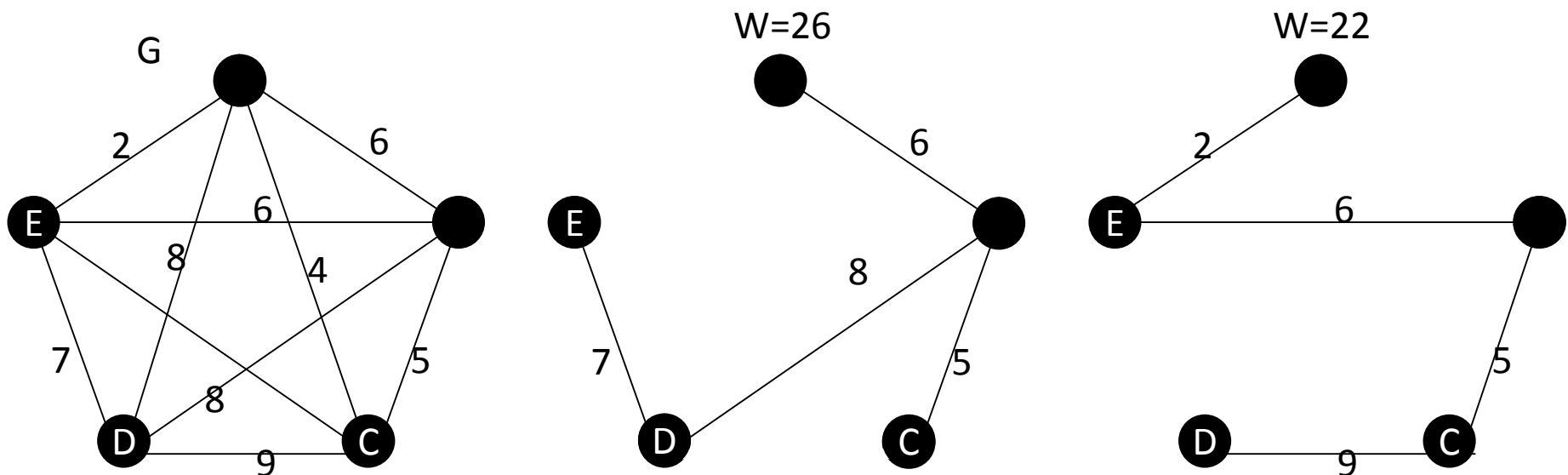
Ελάχιστο γεννητικό δέντρο (Minimum Spanning Tree)

- Ένα ελάχιστο γεννητικό δέντρο για ένα γράφημα G με βάρη στις ακμές είναι
 - ένα υπογράφημα του G που είναι δέντρο (δηλ., είναι συνεκτικό και δεν περιέχει κύκλους),
 - περιέχει όλες τις κορυφές του G και
 - έχει ελάχιστο βάρος



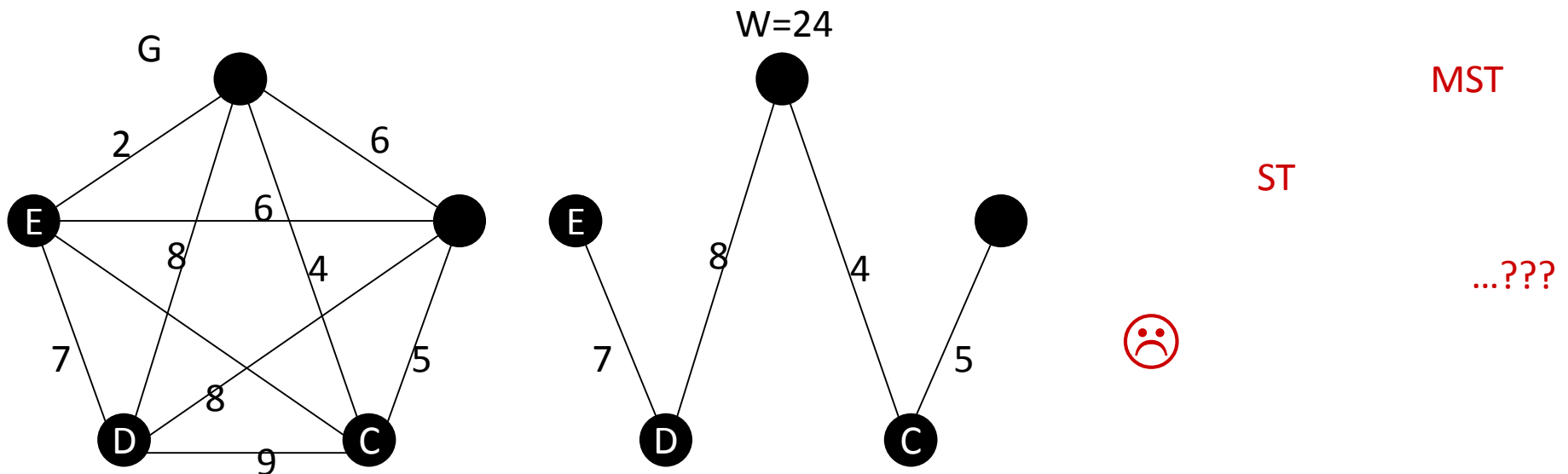
Ελάχιστο γεννητικό δέντρο (Minimum Spanning Tree)

- Ένα ελάχιστο γεννητικό δέντρο για ένα γράφημα G με βάρη στις ακμές είναι
 - ένα υπογράφημα του G που είναι δέντρο (δηλ., είναι συνεκτικό και δεν περιέχει κύκλους),
 - περιέχει όλες τις κορυφές του G και
 - έχει ελάχιστο βάρος



Ελάχιστο γεννητικό δέντρο (Minimum Spanning Tree)

- Ένα ελάχιστο γεννητικό δέντρο για ένα γράφημα G με βάρη στις ακμές είναι
 - ένα υπογράφημα του G που είναι δέντρο (δηλ., είναι συνεκτικό και δεν περιέχει κύκλους),
 - περιέχει όλες τις κορυφές του G και
 - έχει ελάχιστο βάρος



Το πρόβλημα εύρεσης ελάχιστου γεννητικού δένδρου

- MST: Minimum Spanning Tree

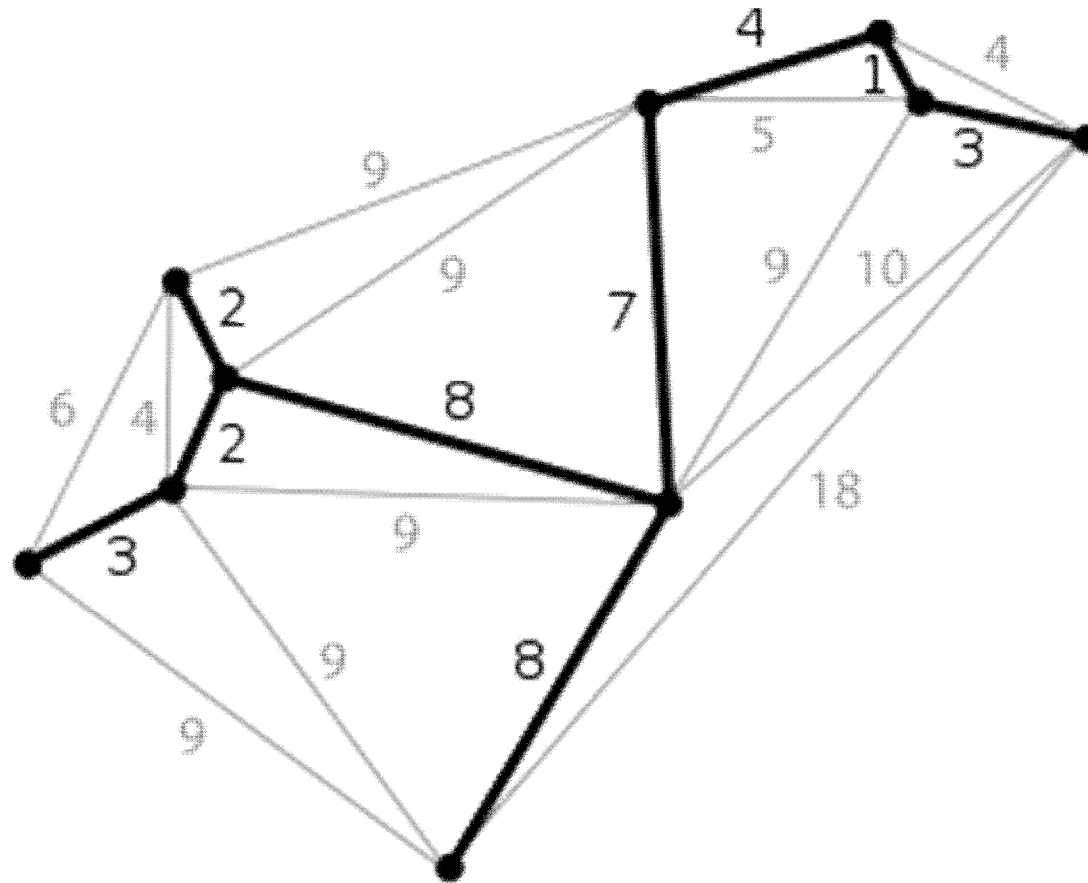
Δεδομένα:

- Συνεκτικό, μη κατευθυνόμενο γράφημα, με βάρη στις ακμές

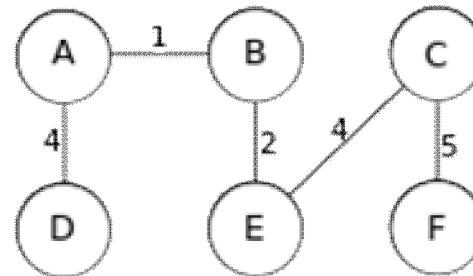
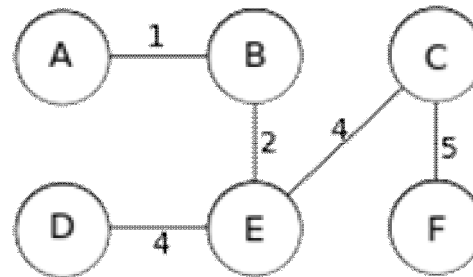
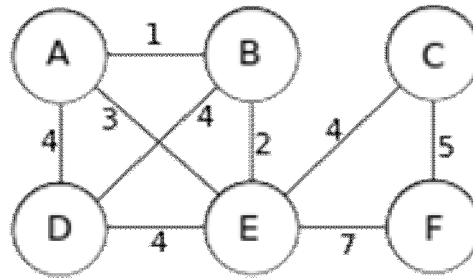
Ζητούμενο:

- Υπογράφημα χωρίς κύκλους (δηλ., **δέντρο**) που συνδέει όλες τις κορυφές (= **γεννητικό**) και έχει **ελάχιστο** βάρος
 - Βάρος υπογραφήματος = άθροισμα βαρών των ακμών του
 - Ένα γράφημα μπορεί να έχει πολλά γεννητικά δέντρα που το καθένα έχει διαφορετικό βάρος
 - Ένα ελάχιστο γεννητικό δέντρο ενός γραφήματος έχει μικρότερο βάρος από κάθε άλλο γεννητικό δέντρο για το γράφημα αυτό

Το πρόβλημα εύρεσης ελάχιστου γεννητικού δένδρου

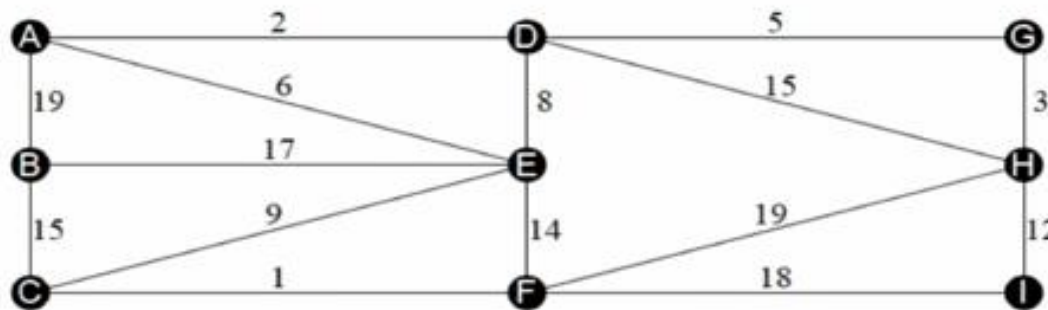
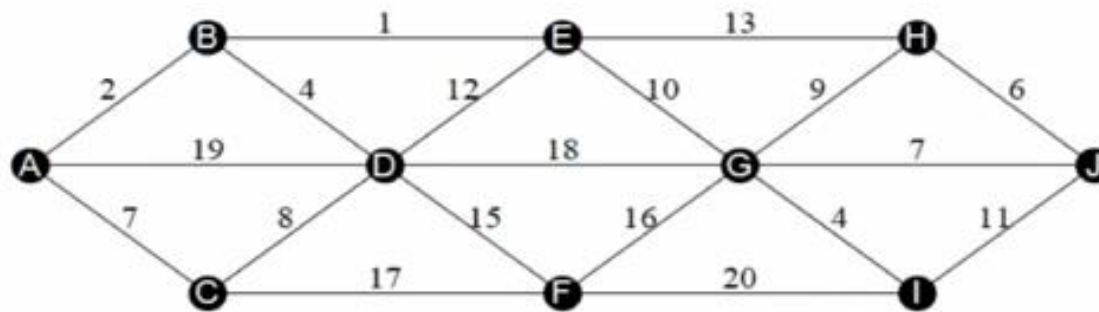


Το πρόβλημα εύρεσης ελάχιστου γεννητικού δένδρου



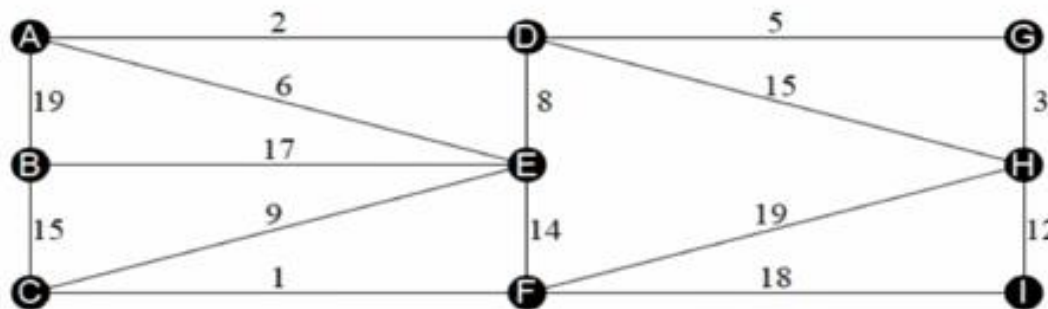
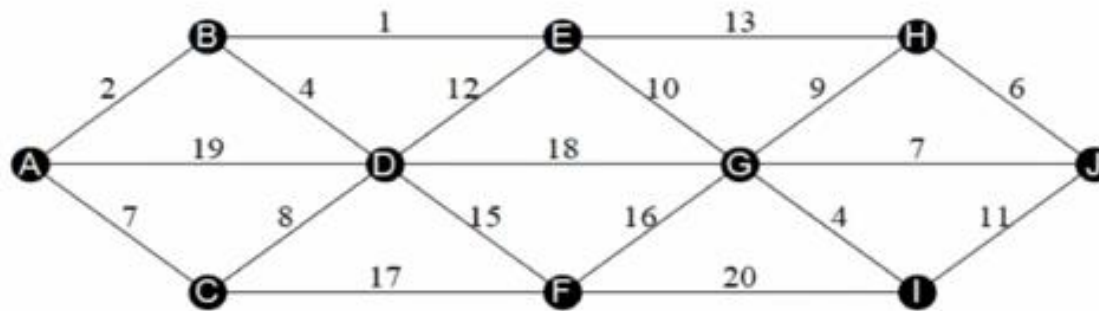
Αλγόριθμοι εύρεσης MST: πρακτικό ενδιαφέρον

- Σενάριο 1: Οι κορυφές αντιστοιχούν σε περιοχές σε αρχαιολογικό χώρο και τα βάρη σε αποστάσεις...
- Ζητούμενο: να περάσουν οι επισκέπτες από όλες τις περιοχές περπατώντας όσο το δυνατόν λιγότερο...



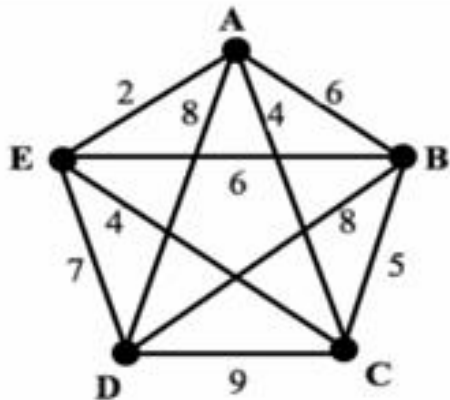
Αλγόριθμοι εύρεσης MST: πρακτικό ενδιαφέρον

- Σενάριο 2: Οι κορυφές αντιστοιχούν σε αισθητήρες που συλλέγουν πληροφορίες σε κάποιο χώρο και τα βάρη στην ακτίνα εκπομπής τους...
- Ζητούμενο: να κρατήσω ενεργοποιημένους κάποιους από τους αισθητήρες ώστε να υπάρχει συνεκτικότητα στο WSN και να ελαχιστοποιείται η συνολική ενέργεια που καταναλώνεται...



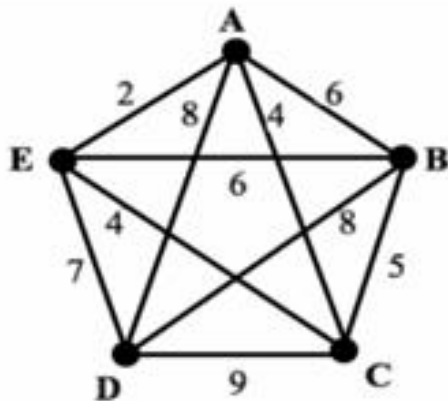
Ο αλγόριθμος του Kruskal

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάταξε όλες τις ακμές σε αύξουσα σειρά ως προς το βάρος τους
 - Βήμα 2: Διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν περισσότερες από μία, διάλεξε τυχαία)
 - Βήμα 3: Από τις ακμές που μένουν, διάλεξε αυτή με το μικρότερο βάρος που δε σχηματίζει κύκλο και πρόσθεσέ την στο T
 - Βήμα 4: Επανάλαβε το Βήμα 3 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ. μέχρι να συνδεθούν όλες οι κορυφές



Ο αλγόριθμος του Kruskal

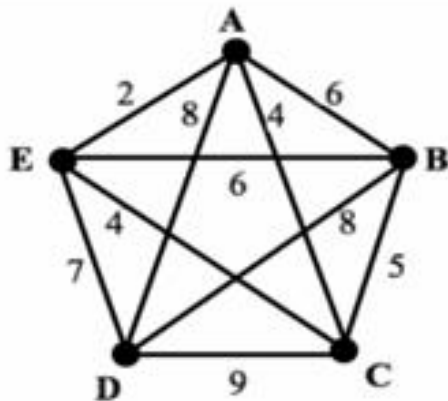
- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάταξε όλες τις ακμές σε αύξουσα σειρά ως προς το βάρος τους
 - Βήμα 2: Διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν περισσότερες από μία, διάλεξε τυχαία)
 - Βήμα 3: Από τις ακμές που μένουν, διάλεξε αυτή με το μικρότερο βάρος που δε σχηματίζει κύκλο και πρόσθεσέ την στο T
 - Βήμα 4: Επανάλαβε το Βήμα 3 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ. μέχρι να συνδεθούν όλες οι κορυφές



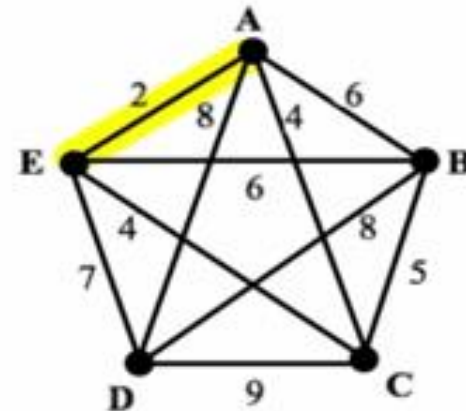
=2
EC=4
AC=4
BC=5
AB=6
BE=6
DE=7
AD=8
BD=8
CD=9

Ο αλγόριθμος του Kruskal

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάταξε όλες τις ακμές σε αύξουσα σειρά ως προς το βάρος τους
 - Βήμα 2: Διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν περισσότερες από μία, διάλεξε τυχαία)
 - Βήμα 3: Από τις ακμές που μένουν, διάλεξε αυτή με το μικρότερο βάρος που δε σχηματίζει κύκλο και πρόσθεσέ την στο T
 - Βήμα 4: Επανάλαβε το Βήμα 3 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ. μέχρι να συνδεθούν όλες οι κορυφές

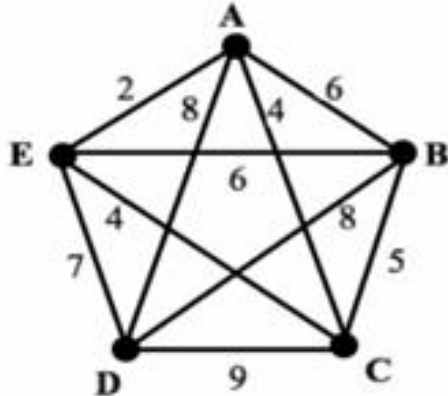


=2 (1)
EC=4
AC=4
BC=5
AB=6
BE=6
DE=7
AD=8
BD=8
CD=9

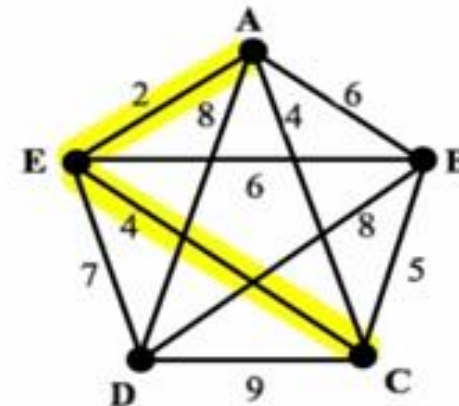


Ο αλγόριθμος του Kruskal

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάταξε όλες τις ακμές σε αύξουσα σειρά ως προς το βάρος τους
 - Βήμα 2: Διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν περισσότερες από μία, διάλεξε τυχαία)
 - Βήμα 3: Από τις ακμές που μένουν, διάλεξε αυτή με το μικρότερο βάρος που δε σχηματίζει κύκλο και πρόσθεσέ την στο T
 - Βήμα 4: Επανάλαβε το Βήμα 3 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ. μέχρι να συνδεθούν όλες οι κορυφές

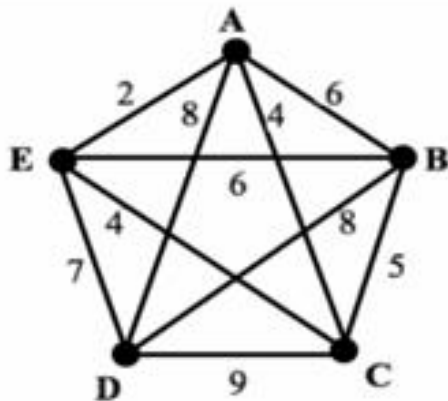


=2 (1)
EC=4 (2)
AC=4
BC=5
AB=6
BE=6
DE=7
AD=8
BD=8
CD=9

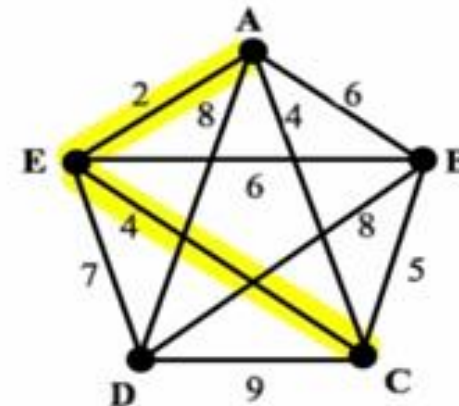


Ο αλγόριθμος του Kruskal

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάταξε όλες τις ακμές σε αύξουσα σειρά ως προς το βάρος τους
 - Βήμα 2: Διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν περισσότερες από μία, διάλεξε τυχαία)
 - Βήμα 3: Από τις ακμές που μένουν, διάλεξε αυτή με το μικρότερο βάρος που δε σχηματίζει κύκλο και πρόσθεσέ την στο T
 - Βήμα 4: Επανάλαβε το Βήμα 3 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ. μέχρι να συνδεθούν όλες οι κορυφές

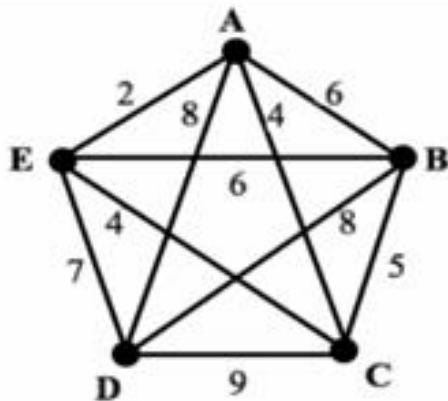


$AE=2$ (1)
 $EC=4$ (2)
 ~~$AC=4$ ()~~
 $BC=5$
 $AB=6$
 $BE=6$
 $DE=7$
 $AD=8$
 $BD=8$
 $CD=9$

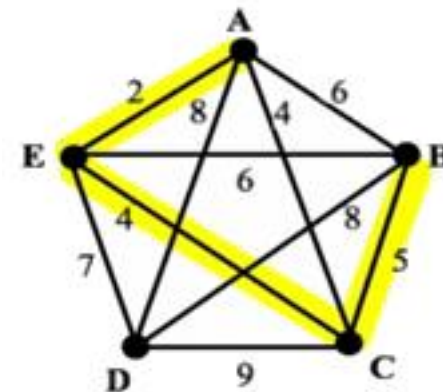


Ο αλγόριθμος του Kruskal

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάταξε όλες τις ακμές σε αύξουσα σειρά ως προς το βάρος τους
 - Βήμα 2: Διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν περισσότερες από μία, διάλεξε τυχαία)
 - Βήμα 3: Από τις ακμές που μένουν, διάλεξε αυτή με το μικρότερο βάρος που δε σχηματίζει κύκλο και πρόσθεσέ την στο T
 - Βήμα 4: Επανάλαβε το Βήμα 3 μέχρι να υπάρχουν n-1 ακμές στο T δηλ. μέχρι να συνδεθούν όλες οι κορυφές

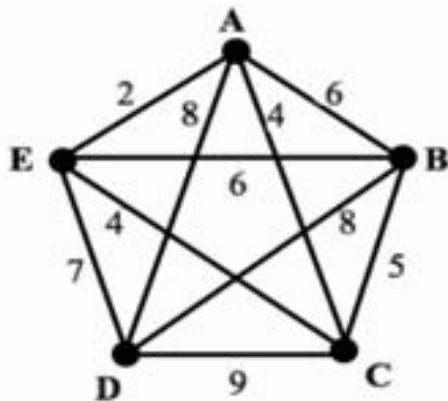


=2 (1)
EC=4 (2)
~~AC=4 ()~~
BC=5 (3)
AB=6
BE=6
DE=7
AD=8
BD=8
CD=9

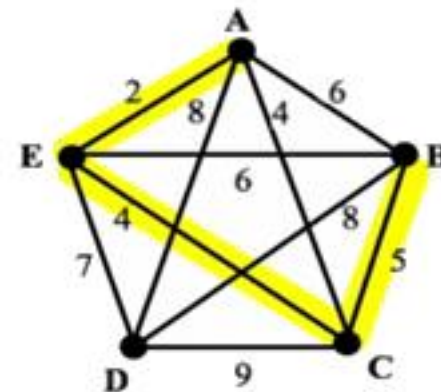


Ο αλγόριθμος του Kruskal

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάταξε όλες τις ακμές σε αύξουσα σειρά ως προς το βάρος τους
 - Βήμα 2: Διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν περισσότερες από μία, διάλεξε τυχαία)
 - Βήμα 3: Από τις ακμές που μένουν, διάλεξε αυτή με το μικρότερο βάρος που δε σχηματίζει κύκλο και πρόσθεσέ την στο T
 - Βήμα 4: Επανάλαβε το Βήμα 3 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ. μέχρι να συνδεθούν όλες οι κορυφές

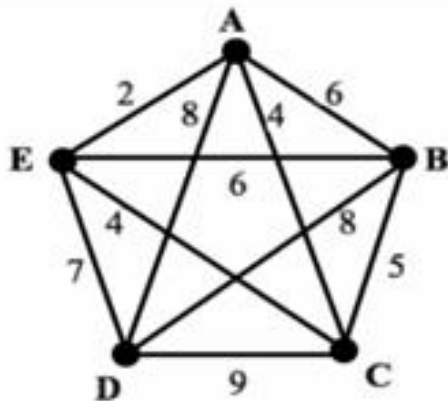


$AE=2$ (1)
 $EC=4$ (2)
 ~~$AC=4$ ()~~
 $BC=5$ (3)
 ~~$AB=6$ ()~~
 ~~$BE=6$ ()~~
 $DE=7$
 $AD=8$
 $BD=8$
 $CD=9$

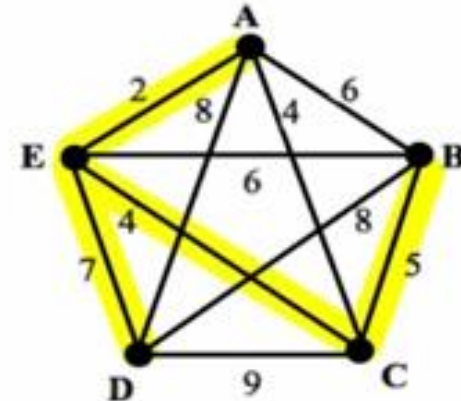


Ο αλγόριθμος του Kruskal

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάταξε όλες τις ακμές σε αύξουσα σειρά ως προς το βάρος τους
 - Βήμα 2: Διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν περισσότερες από μία, διάλεξε τυχαία)
 - Βήμα 3: Από τις ακμές που μένουν, διάλεξε αυτή με το μικρότερο βάρος που δε σχηματίζει κύκλο και πρόσθεσέ την στο T
 - Βήμα 4: Επανάλαβε το Βήμα 3 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ. μέχρι να συνδεθούν όλες οι κορυφές

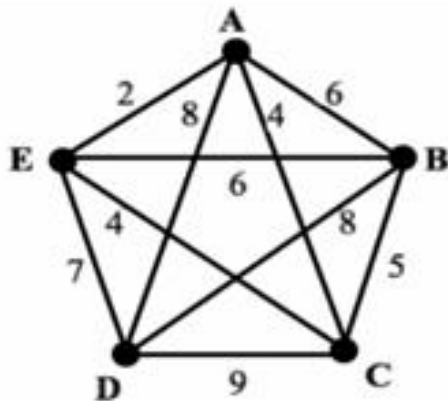


AE=2 (1)
EC=4 (2)
~~AC=4 ()~~
BC=5 (3)
~~AB=6 ()~~
~~BE=6 ()~~
DE=7 (4)
AD=8
BD=8
CD=9

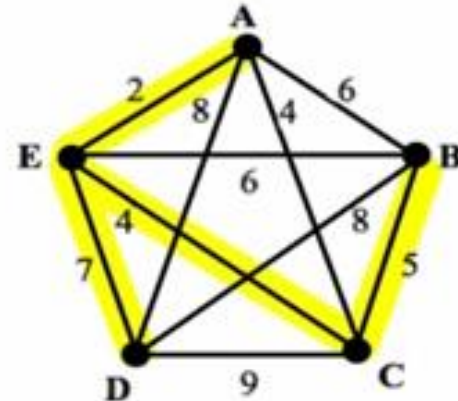


Ο αλγόριθμος του Kruskal

- Ο αλγόριθμος τερματίζει αφού:
 - Έχουμε ST με $n-1$ ($=5-1=4$) ακμές, ή εναλλακτικά,
 - Έχουμε ST που περιέχει όλες τις κορυφές του δοσμένου γραφήματος
 - Παρατηρήστε ότι καμία από τις AD, BD, CD δε μπορεί έτσι κι αλλιώς να προστεθεί στο ST γιατί θα δημιουργούσε κύκλο...
- Το MST που βρήκε περιέχει τις ακμές:
 - AE (2), EC (4), BC (5), DE (=7) και έχει συνολικό βάρος $2+4+5+7=18$

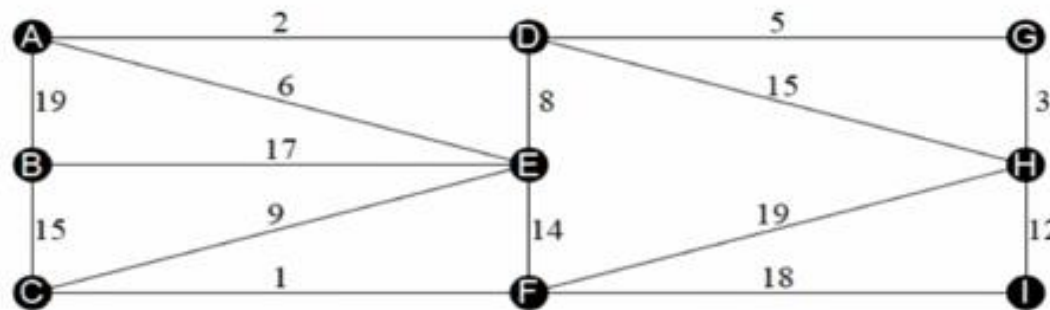
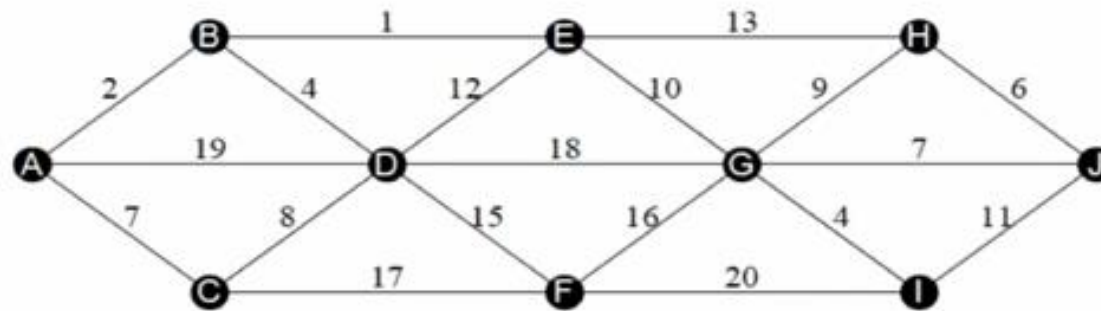


AE=2 (1)
EC=4 (2)
~~AC=4~~ ()
BC=5 (3)
~~AB=6~~ ()
~~BE=6~~ ()
DE=7 (4)
AD=8
BD=8
CD=9



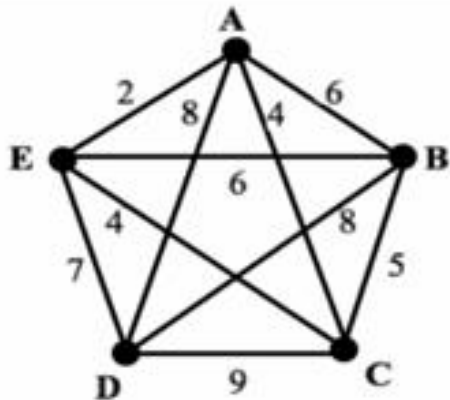
Ο αλγόριθμος του Kruskal: εξάσκηση

- Εκτελέστε τον αλγόριθμο του Kruskal για την εύρεση MST στα δύο γραφήματα που ακολουθούν
- Ποιο είναι το βάρος του MST σε κάθε περίπτωση;



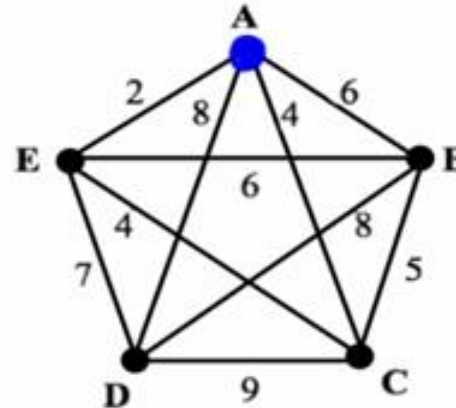
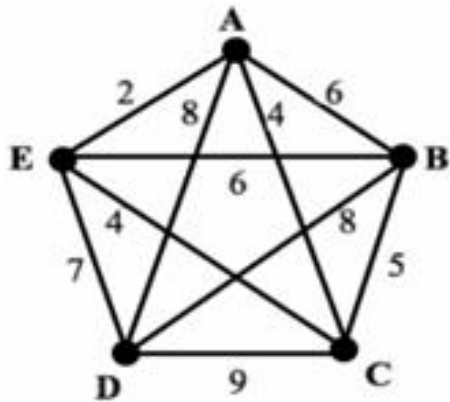
Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές



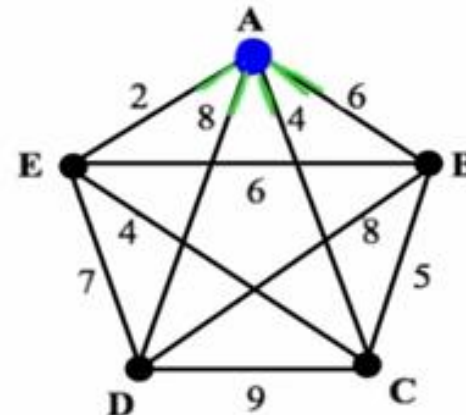
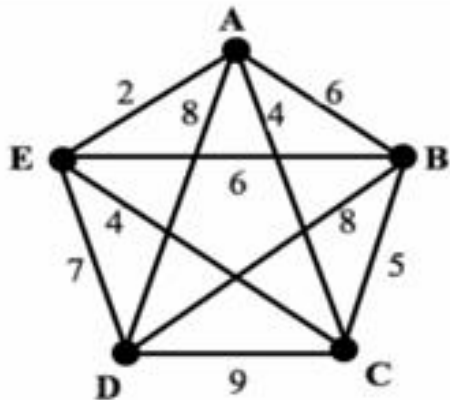
Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές



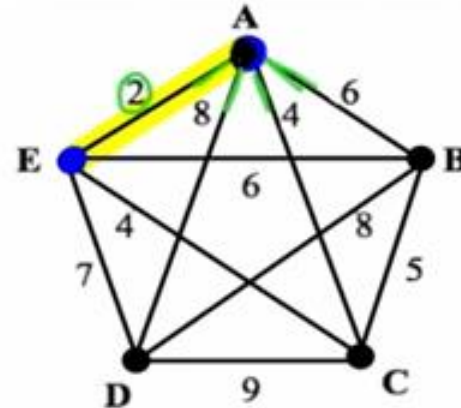
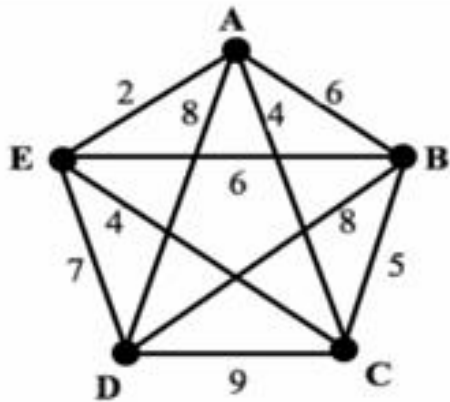
Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές



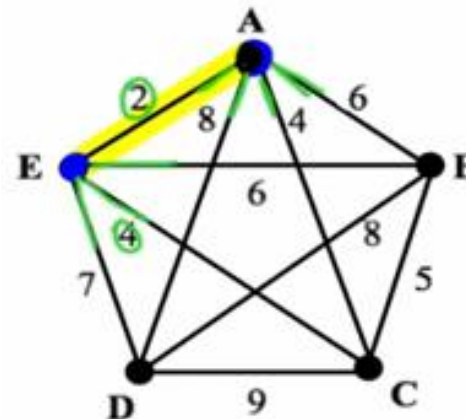
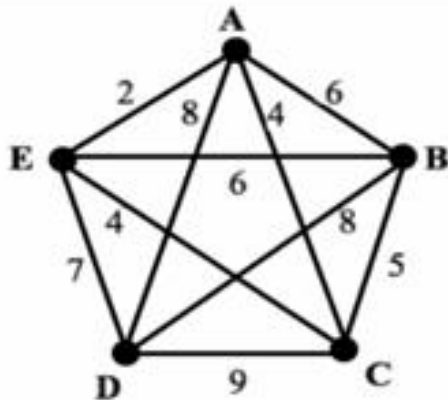
Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές



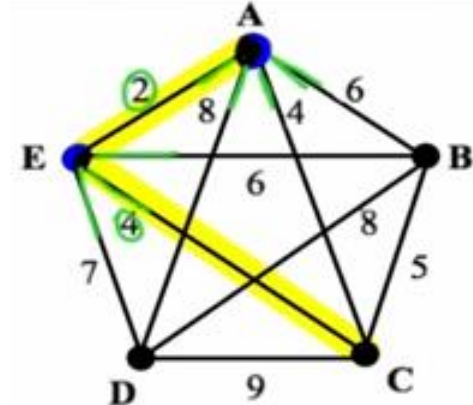
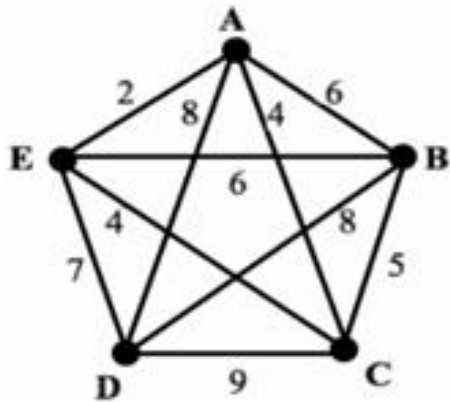
Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές



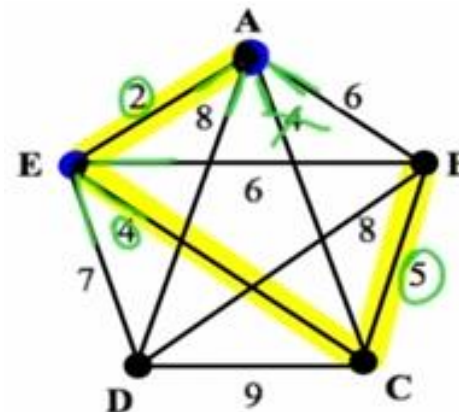
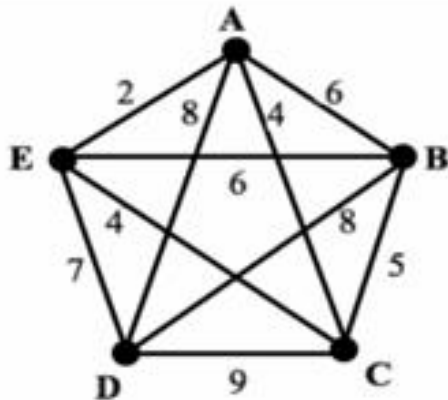
Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές



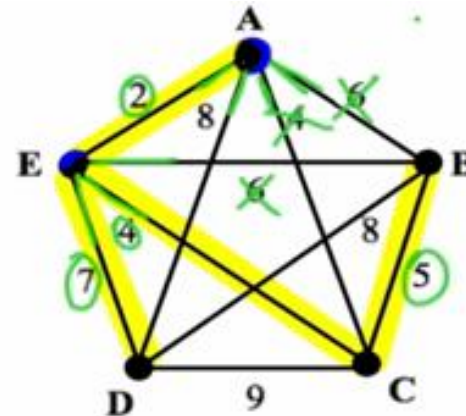
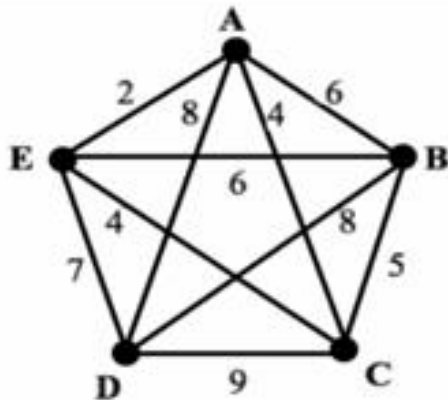
Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές



Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές



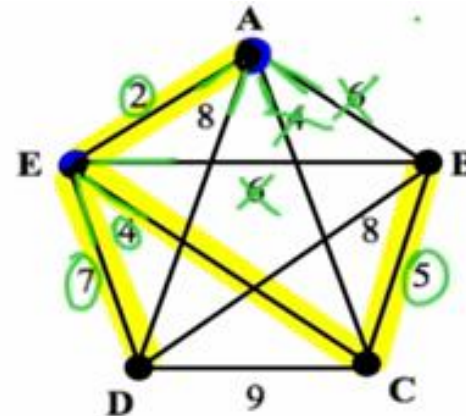
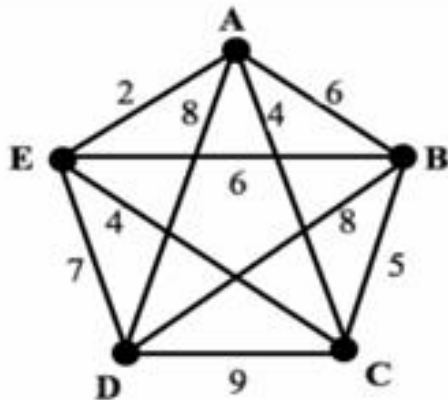
Ο αλγόριθμος του Prim

- Βρίσκει το ελάχιστο γεννητικό δέντρο (MST) T σε δοσμένο γράφημα
 - Βήμα 1: Διάλεξε αυθαίρετη κορυφή να είναι η πρώτη στο δέντρο T
 - Βήμα 2: Δες ποιες ακμές από κορυφές στο T συνδέονται με κορυφές εκτός του T και διάλεξε την ακμή με το μικρότερο βάρος να ανήκει στο T (αν υπάρχουν παραπάνω από μία, διάλεξε τυχαία)
 - Βήμα 3: Επανάλαβε το Βήμα 2 μέχρι να υπάρχουν $n-1$ ακμές στο T δηλ., μέχρι να συνδεθούν όλες οι κορυφές

MST: AE EC BC ED

2 4 5 7

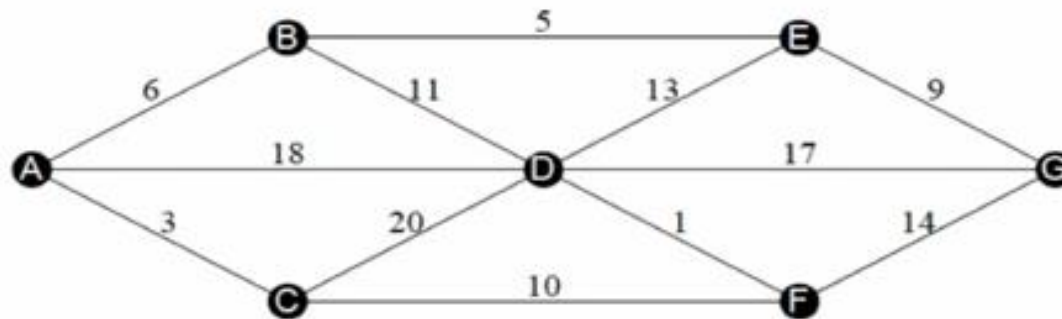
: 18



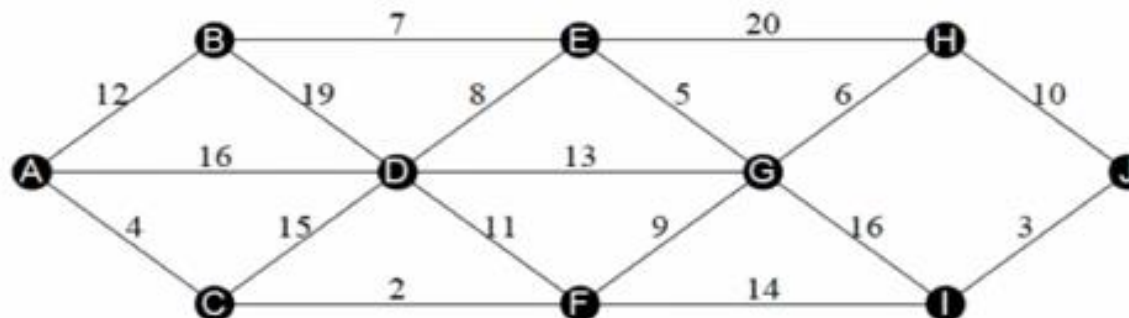
Ο αλγόριθμος του Prim: εξάσκηση

- Εκτελέστε τον αλγόριθμο του Prim για την εύρεση MST στα δύο γραφήματα που ακολουθούν ξεκινώντας από την κορυφή A
- Ποιο είναι το βάρος του MST σε κάθε περίπτωση;

a)



b)



Το πρόβλημα εύρεσης ελάχιστου γεννητικού δένδρου: ο αλγόριθμος του Kruskal

- Άπληστος αλγόριθμος που βρίσκει ένα ελάχιστο γεννητικό δέντρο (MST) σε δοσμένο συνεκτικό γράφημα με βάρη
- Βρίσκει σύνολο από ακμές που σχηματίζουν δέντρο που περιέχει όλες τις κορυφές του γραφήματος και έχει ελάχιστο συνολικό βάρος
- *Ο αλγόριθμος προτάθηκε από τον Joseph Kruskal εμφανίστηκε στα Πρακτικά του American Mathematical Society, σελ. 48–50 το 1956*

Το πρόβλημα εύρεσης ελάχιστου γεννητικού δένδρου: ο αλγόριθμος του Prim

- Άπληστος αλγόριθμος που βρίσκει ένα ελάχιστο γεννητικό δέντρο (MST) σε δοσμένο συνεκτικό γράφημα με βάρη
- Βρίσκει σύνολο από ακμές που σχηματίζουν δέντρο που περιέχει όλες τις κορυφές του γραφήματος και έχει ελάχιστο συνολικό βάρος
- *Ο αλγόριθμος προτάθηκε από τον Τσέχο μαθηματικό Vojtěch Jarník το 1930 και αργότερα – ανεξάρτητα - από τον επιστήμονα υπολογιστών Robert C. Prim το 1957 και ανακαλύφθηκε εκ νέου από τον Edsger Dijkstra το 1959*
 - *V. Jarník: O jistém problému minimálním [About a certain minimal problem], Práce Moravské Přírodovědecké Společnosti, 6, 1930, pp. 57–63.*
 - *R. C. Prim: Shortest connection networks and some generalizations. In: Bell System Technical Journal, 36 (1957), pp. 1389–1401*

Σύγκριση

Αλγόριθμος Kruskal

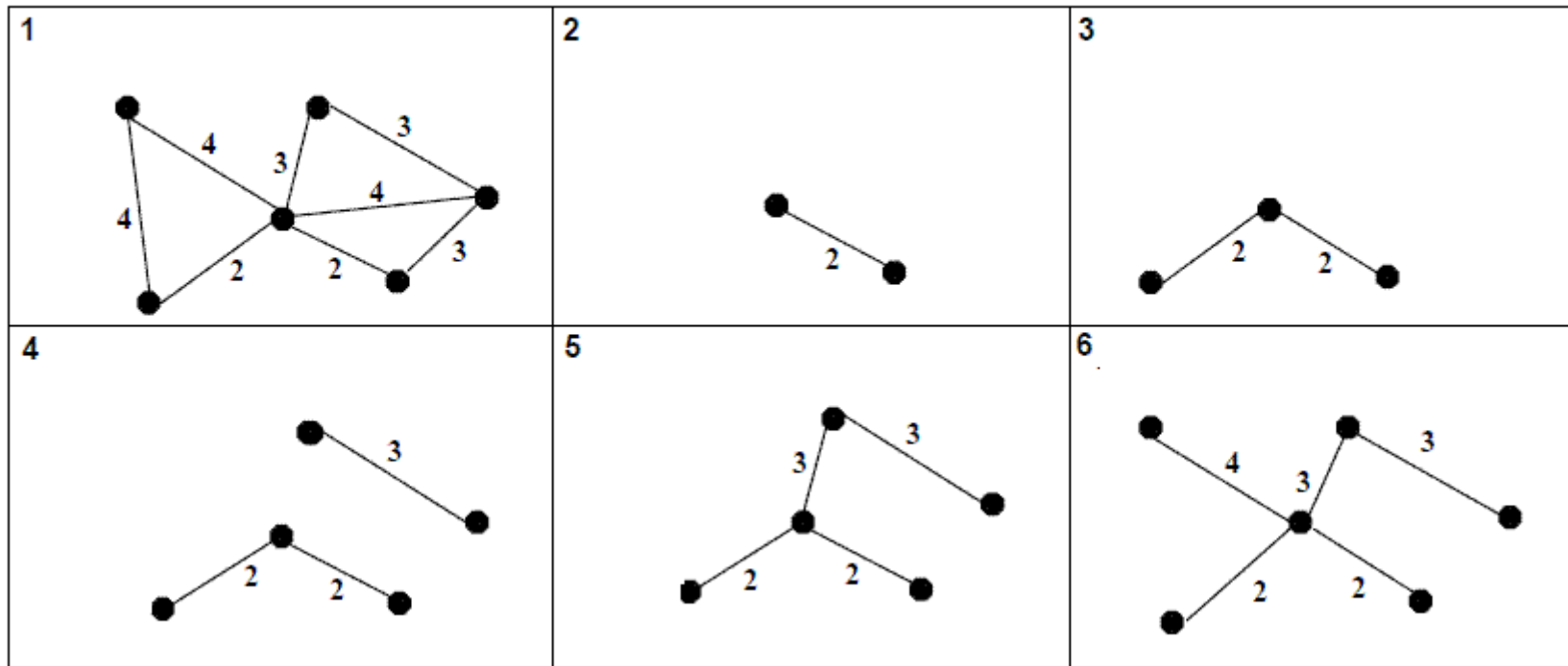
- Ξεκινάει με ακμή
- Διατηρεί δάσος (forest) – όχι συνεκτική συνιστώσα
- Επιλέγει ακμές όχι απαραίτητα γειτονικές
- Εξετάζει κάθε ακμή μία φορά
- Λειτουργεί και σε μη συνεκτικά γραφήματα
- Καλύτερος για αραιά γραφήματα
- $O(n^2 \log n)$ για πυκνά γραφήματα
- $O(n \log n)$ για αραιά γραφήματα

Αλγόριθμος Prim

- Ξεκινάει με κορυφή
- Διατηρεί συνεκτική συνιστώσα
- Δημιουργεί το δέντρο από κορυφή σε κορυφή
- Εξετάζει μία ακμή πολλές φορές
- Το γράφημα πρέπει να είναι συνεκτικό
- Καλύτερος για πυκνά γραφήματα
- $O(n^2)$

Σύγκριση

Αλγόριθμος Kruskal



Σύγκριση

Αλγόριθμος Prim

