



Δομές Δεδομένων

Ενότητα 6: Γράφοι

Απόστολος Παπαδόπουλος
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

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.





Γράφοι



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα ενότητας

1. DFS σε κατευθυνόμενο γράφο
2. Ελάχιστα Μονοπάτια
3. Τοπολογική Ταξινόμηση
4. Ελάχιστα Ζευγνύοντα Δένδρα





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Γράφοι

Depth First Search

- Μέθοδος που χρησιμοποιείται ευρύτατα από πολλούς αλγορίθμους γραφημάτων. Μπορεί να χρησιμοποιηθεί για:
 - Να απαντήσει αν ο γράφος είναι συνεκτικός ή όχι,
 - Να προσδιοριστούν οι συνεκτικές συνιστώσες του γράφου,
 - Να προσδιορίσουμε τα σημεία άρθρωσης,
 - ...



Αλγόριθμος DFS

DFS(G)

```
{  
  
  for each vertex  $u \in G \rightarrow V$   
  {  
     $u \rightarrow \text{color} = \text{EMPTY};$   
  }  
  
  time = 0;  
  for each vertex  $u \in G \rightarrow V$   
  {  
    if ( $u \rightarrow \text{color} == \text{EMPTY}$ )  
      DFS_Visit(u);  
  }  
}
```

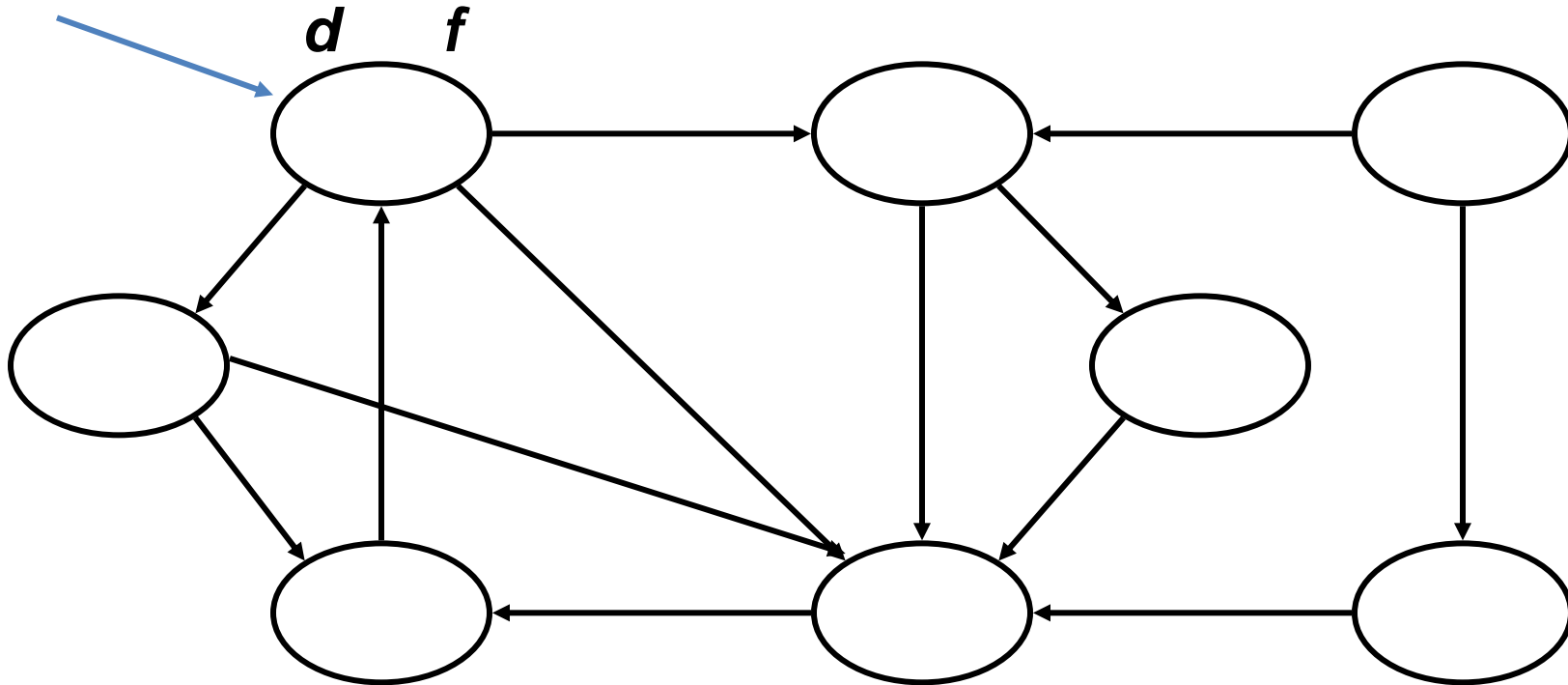
DFS_Visit(u)

```
{  
  
   $u \rightarrow \text{color} = \text{YELLOW};$   
  time = time+1;  
   $u \rightarrow d = \text{time};$   
  for each  $v \in u \rightarrow \text{Adj}[]$   
  {  
    if ( $v \rightarrow \text{color} == \text{EMPTY}$ )  
      DFS_Visit(v);  
  }  
   $u \rightarrow \text{color} = \text{WHITE};$   
  time = time+1;  
   $u \rightarrow f = \text{time};$   
}
```



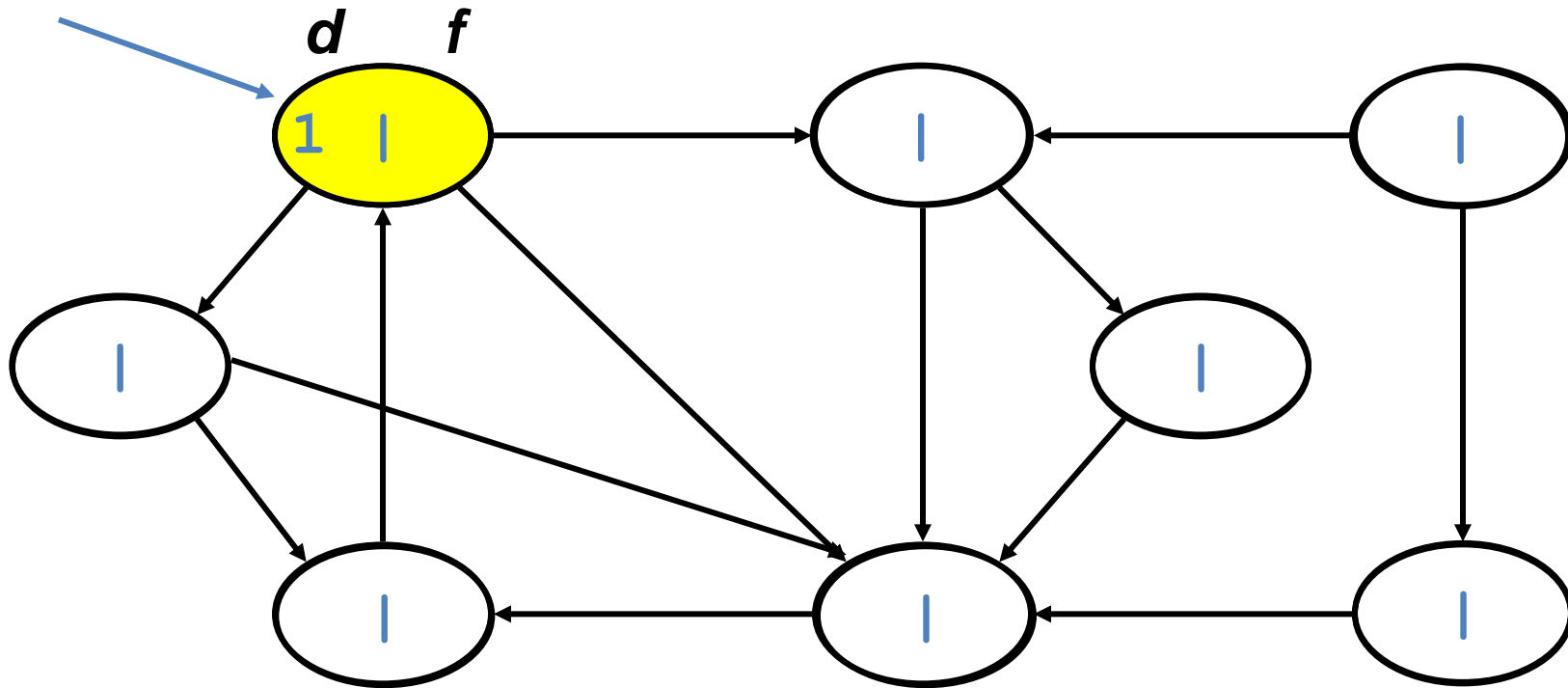
Παράδειγμα DFS

Πηγή



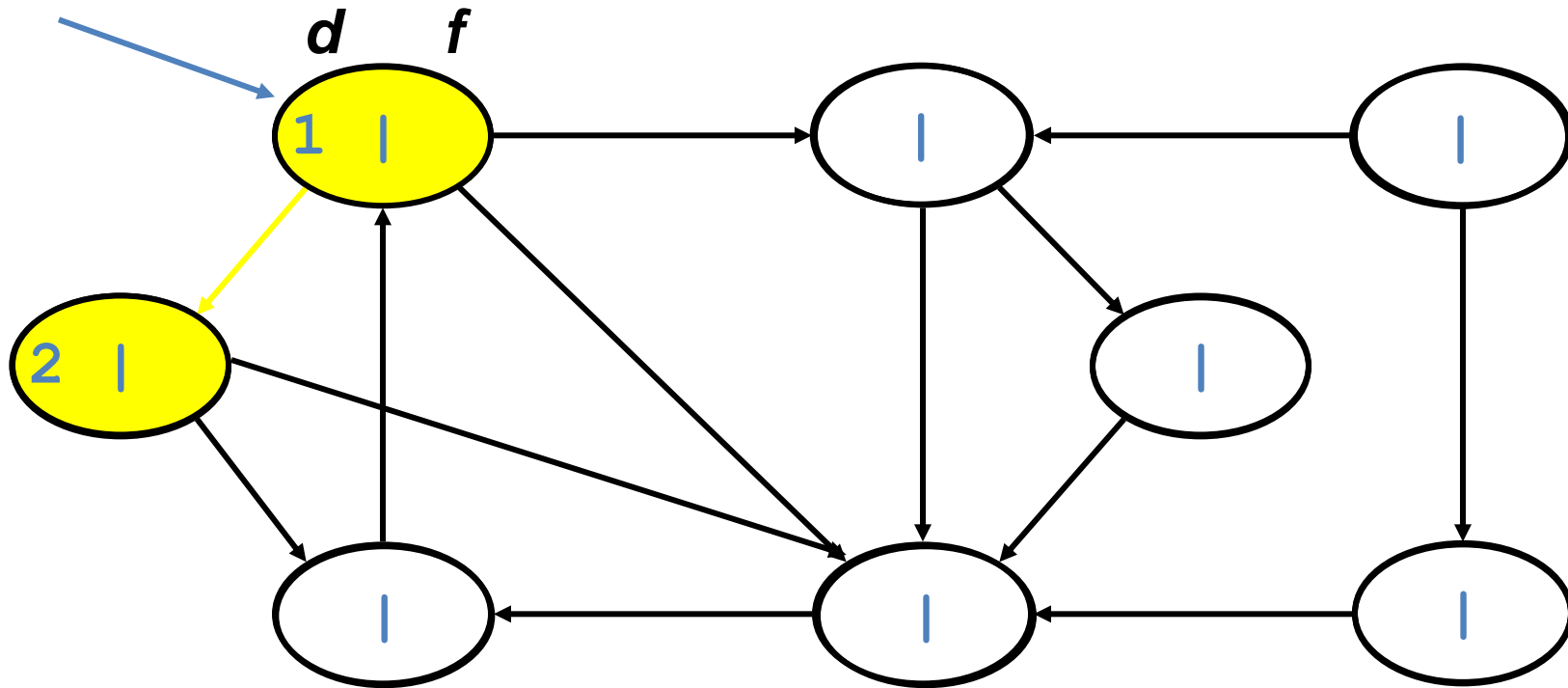
Παράδειγμα DFS

Πηγή



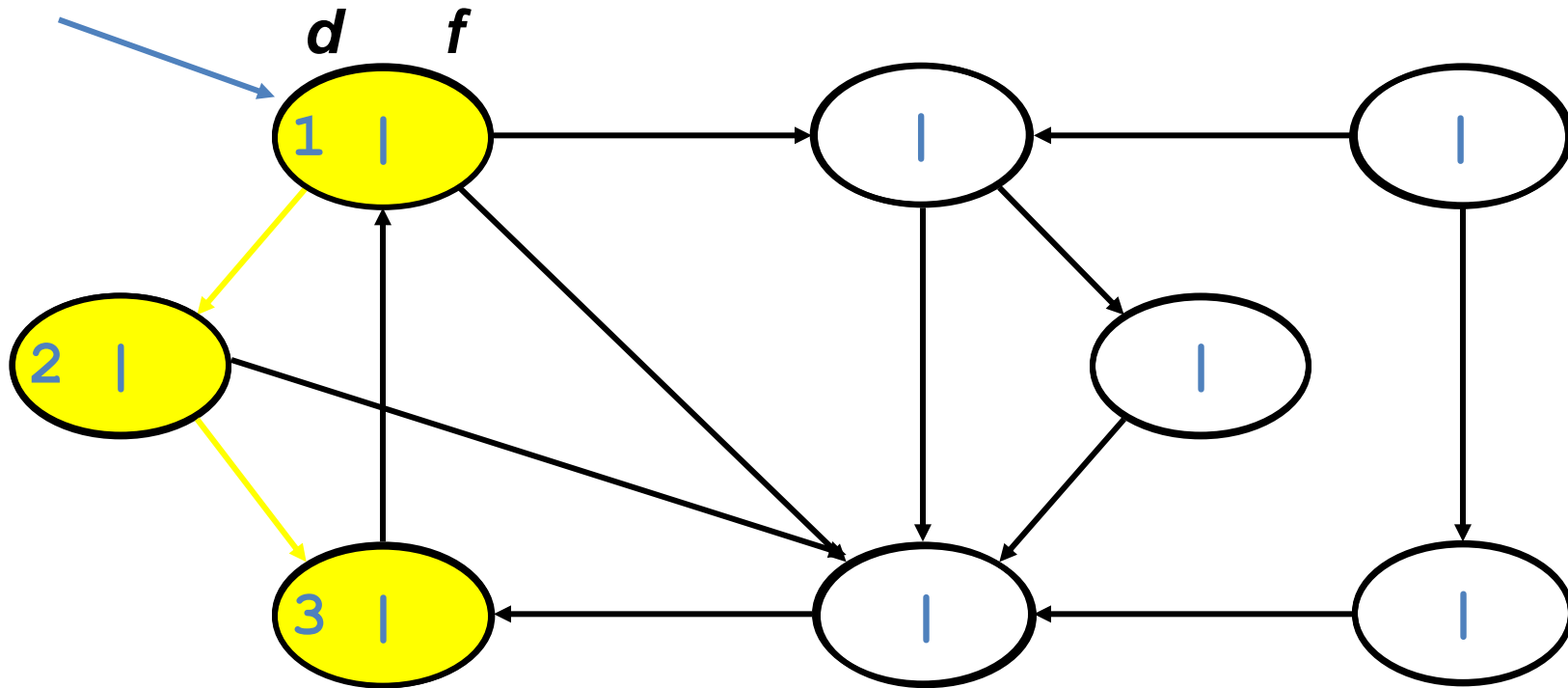
Παράδειγμα DFS

Πηγή



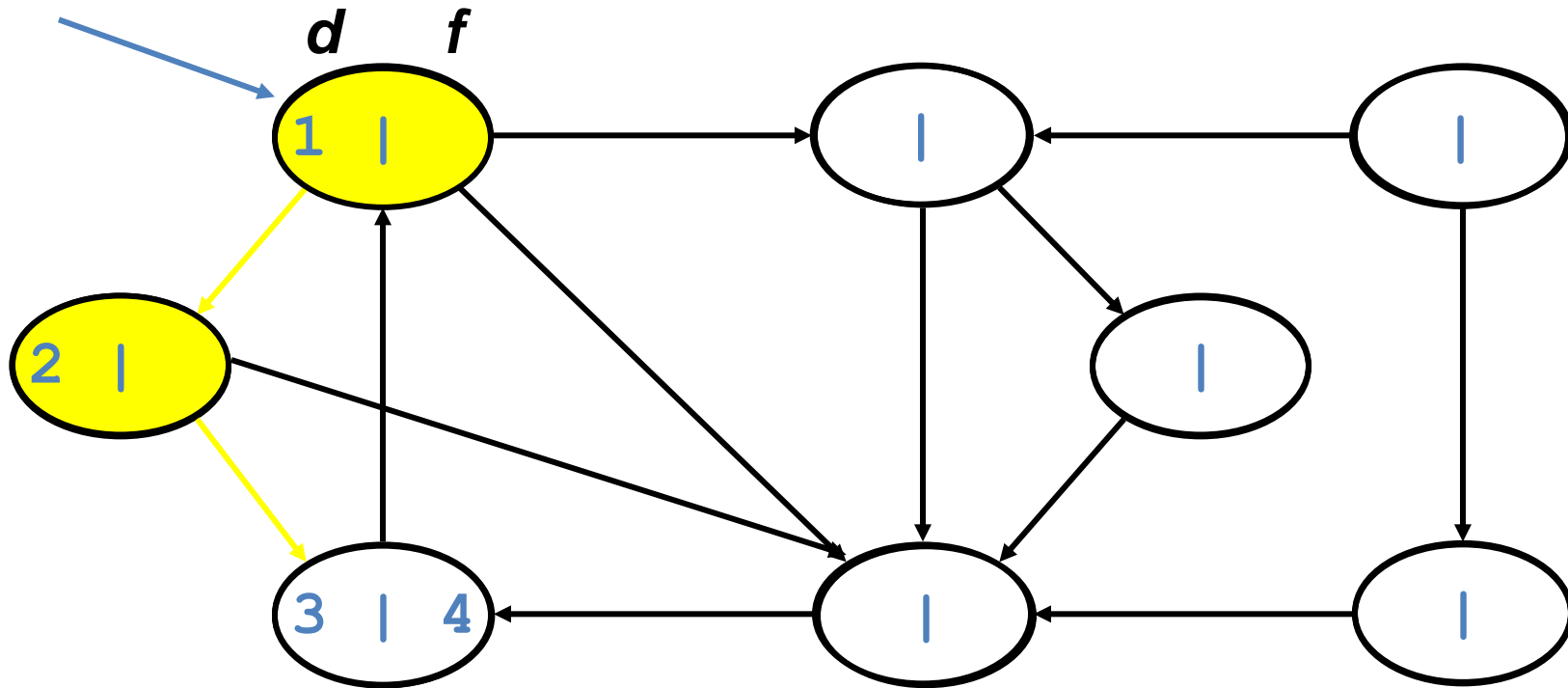
Παράδειγμα DFS

Πηγή



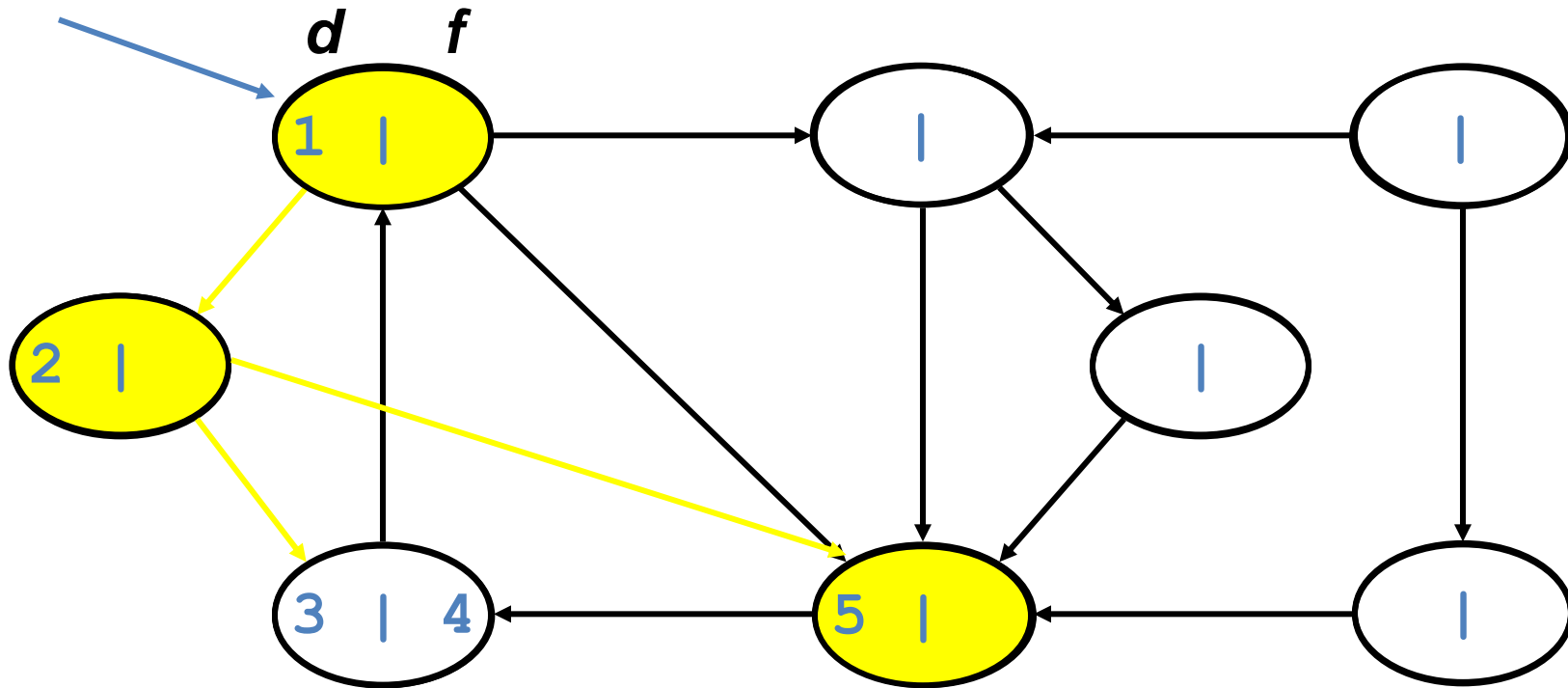
Παράδειγμα DFS

Πηγή



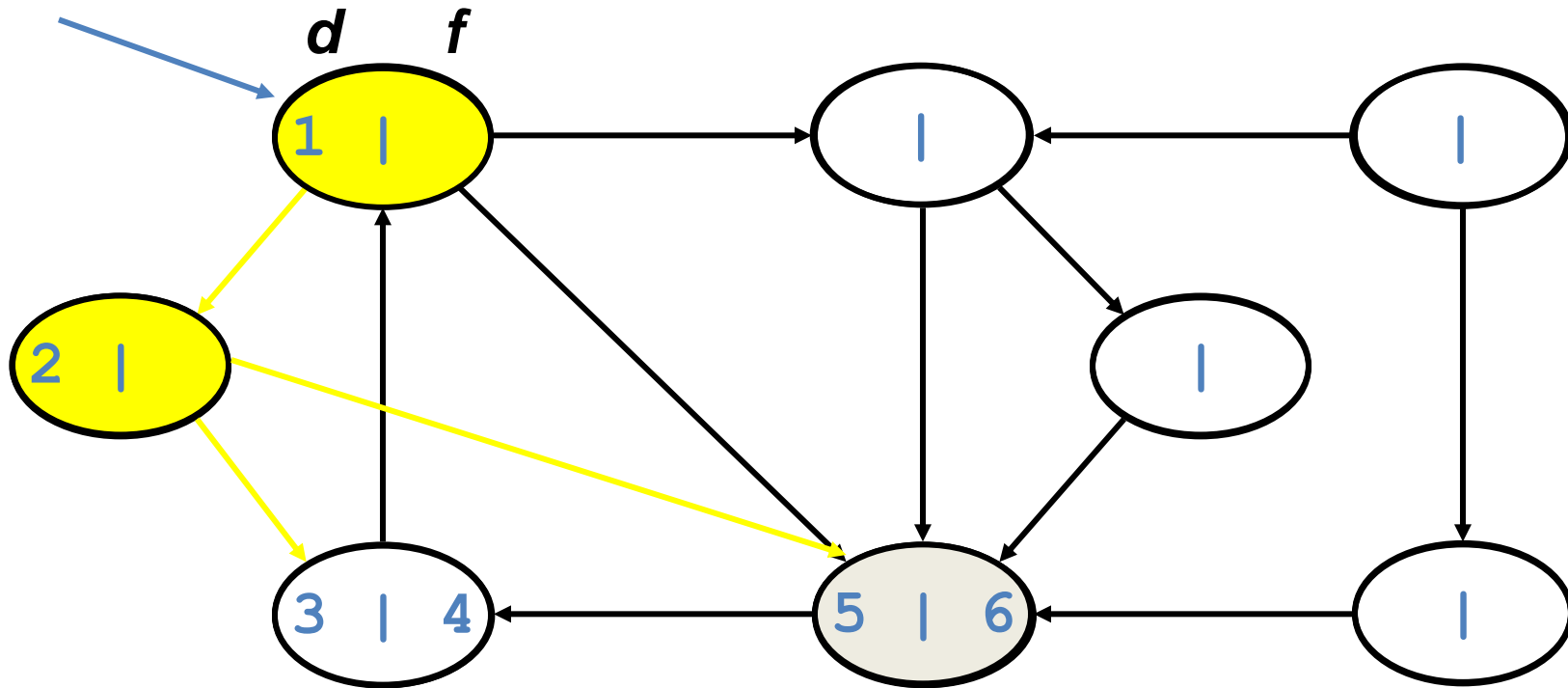
Παράδειγμα DFS

Πηγή



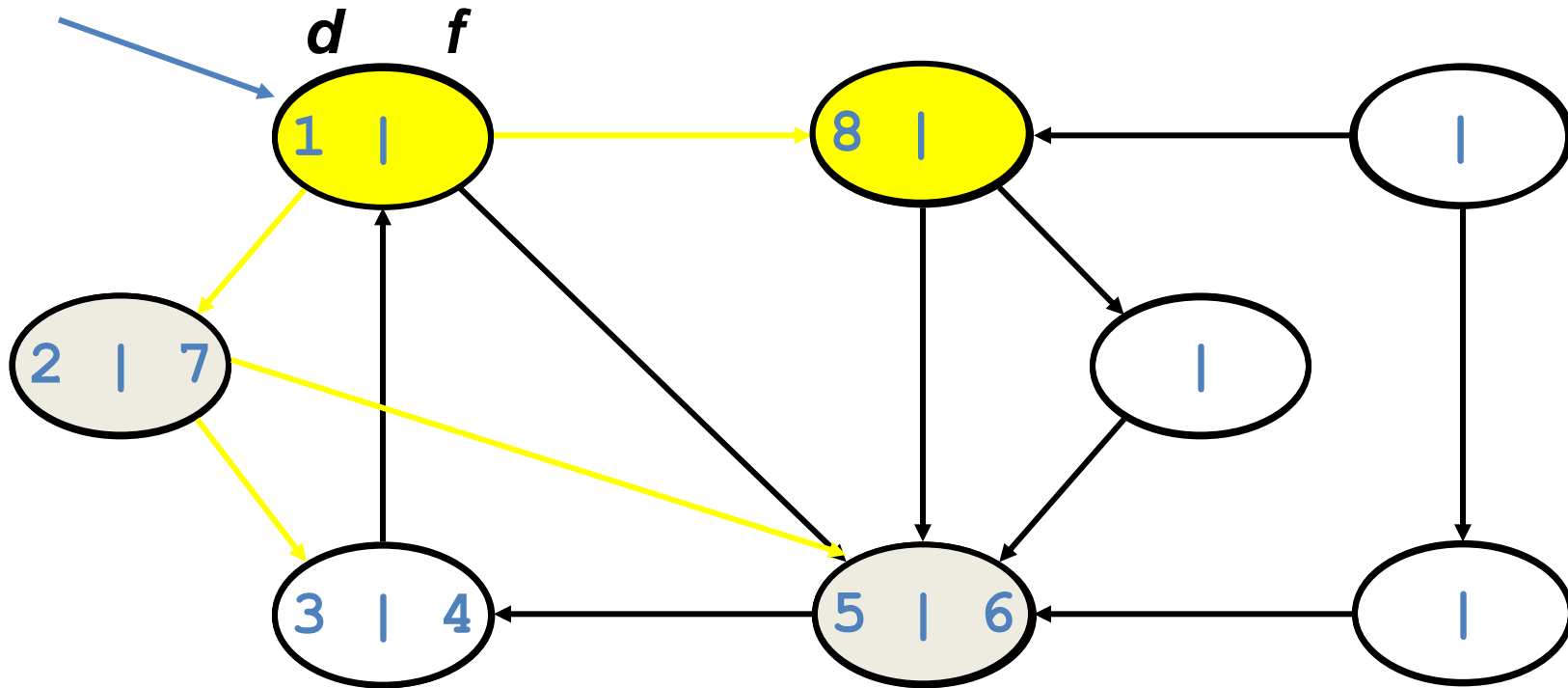
Παράδειγμα DFS

Πηγή



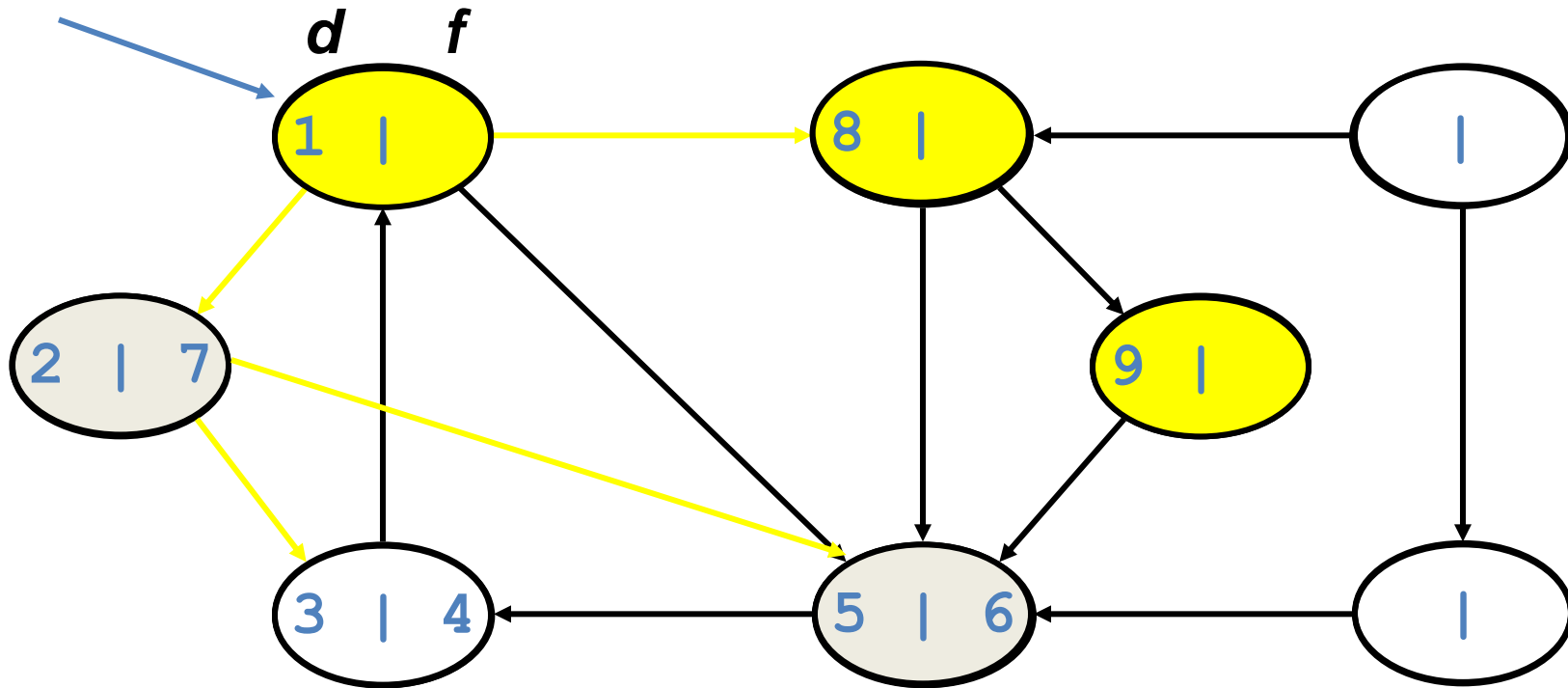
Παράδειγμα DFS

Πηγή



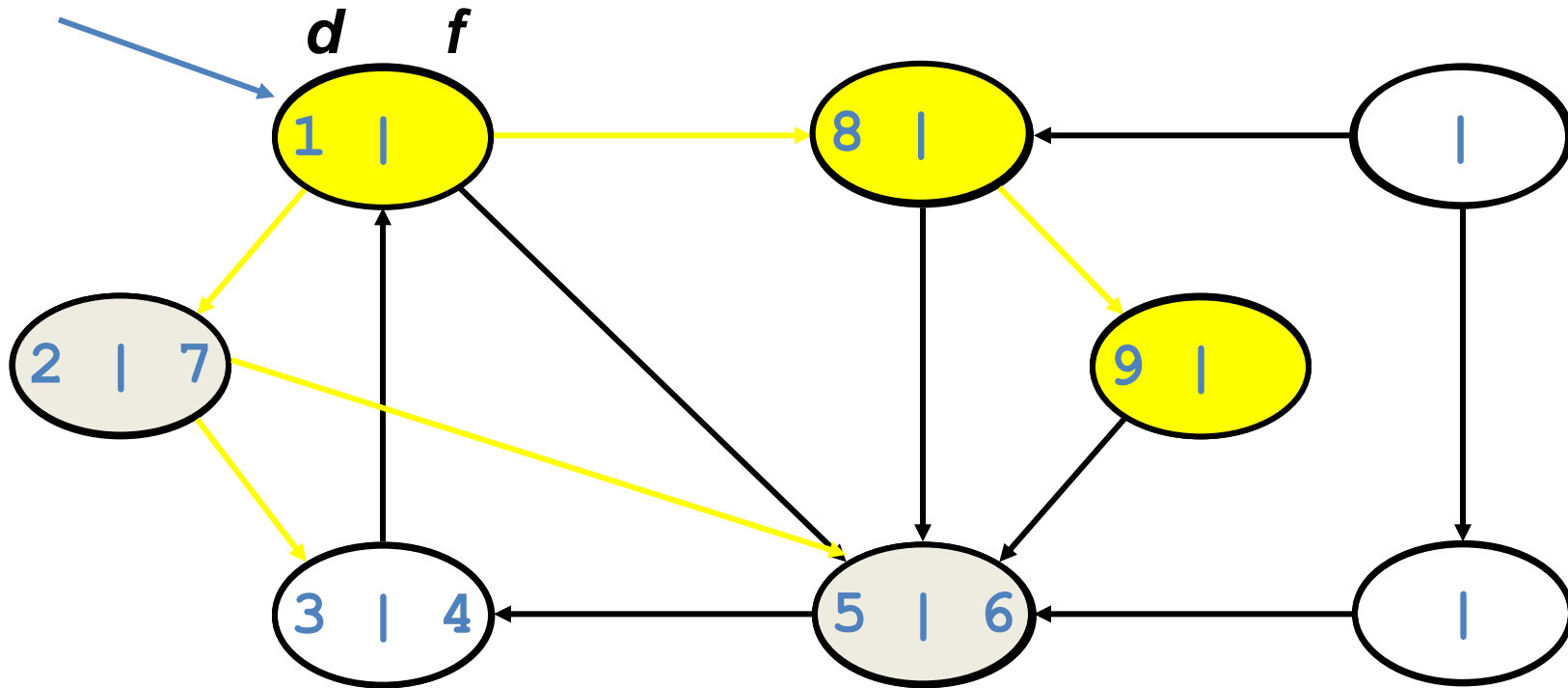
Παράδειγμα DFS

Πηγή



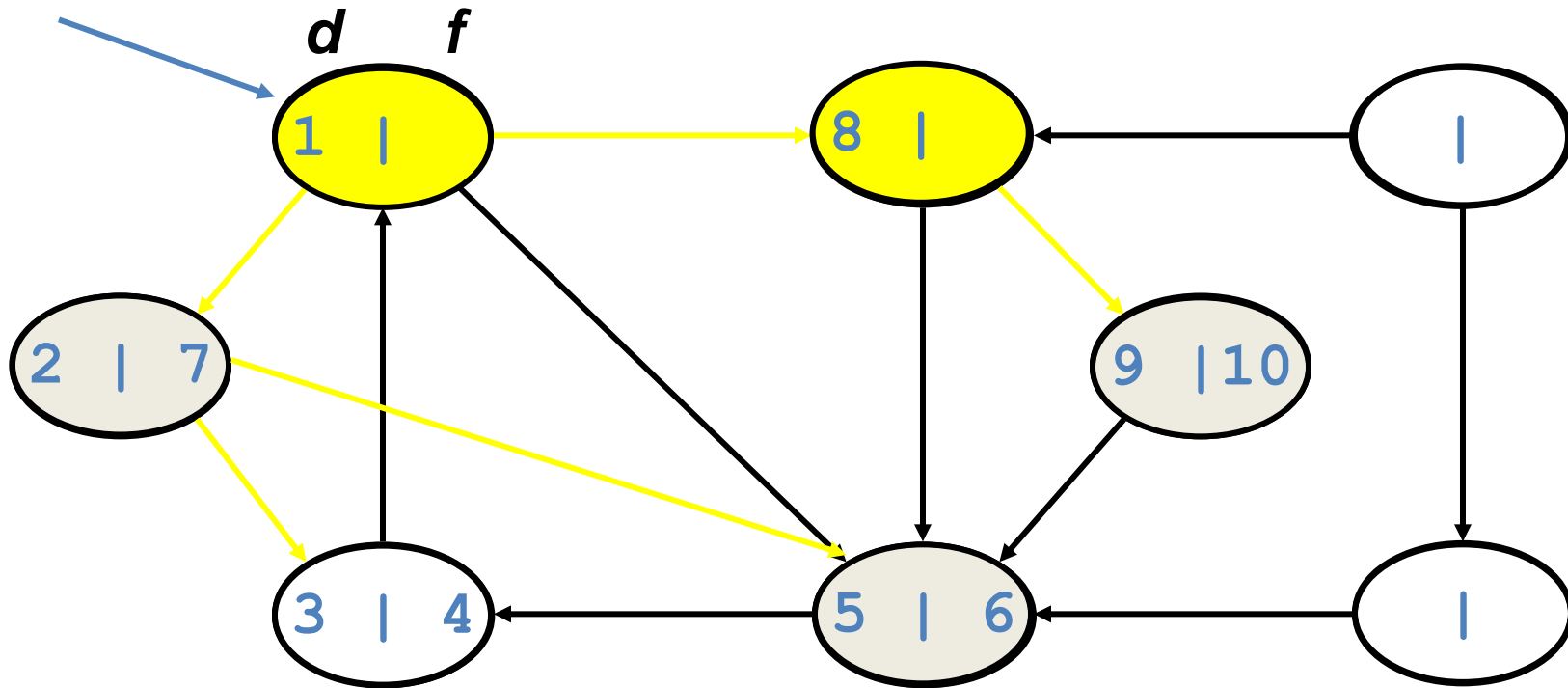
Παράδειγμα DFS

Πηγή



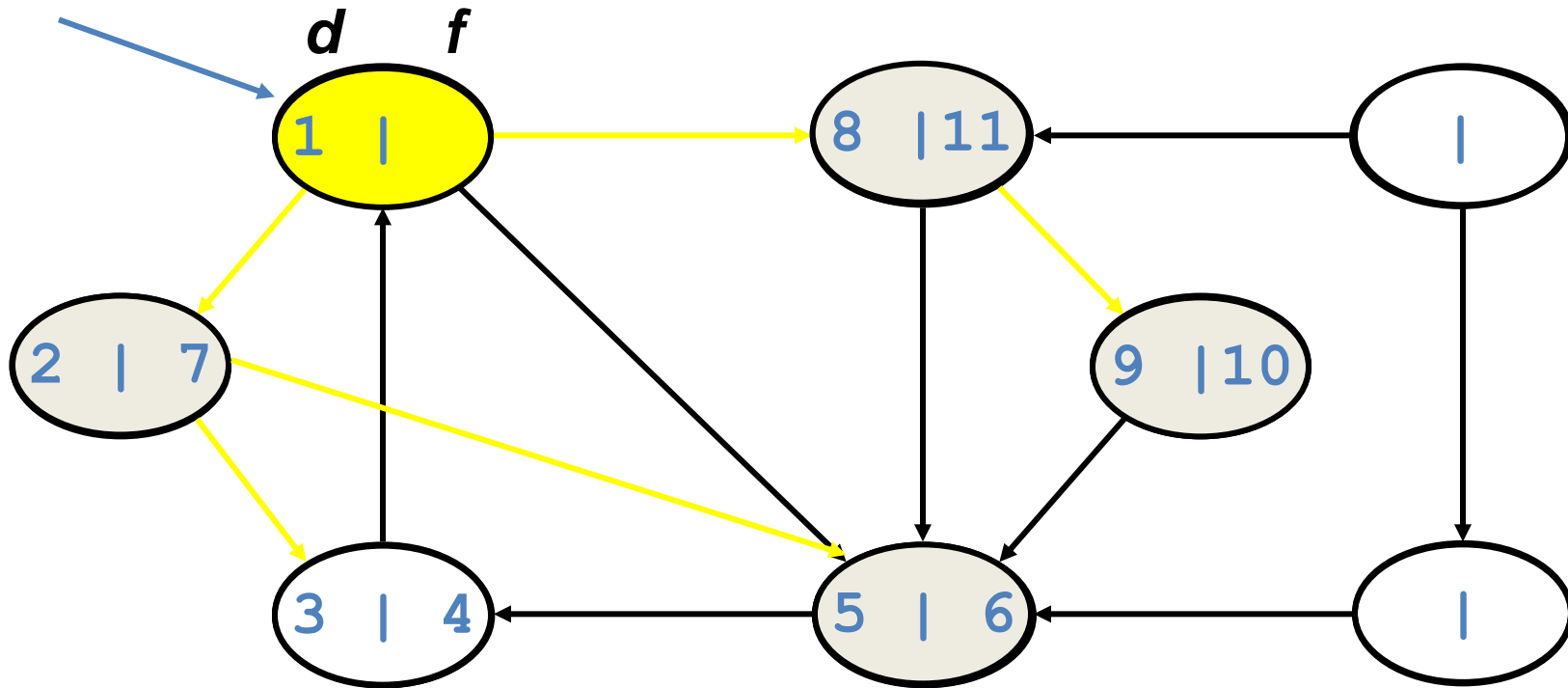
Παράδειγμα DFS

Πηγή



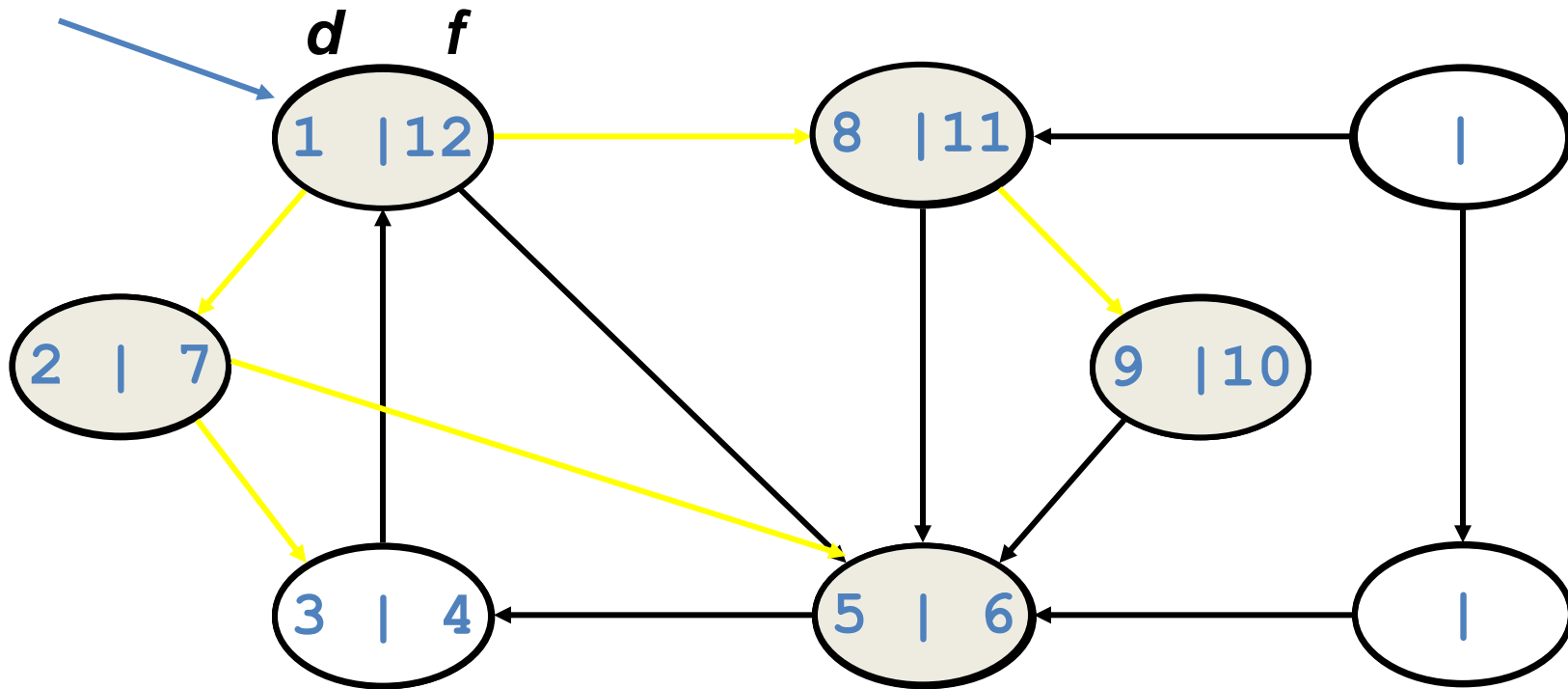
Παράδειγμα DFS

Πηγή



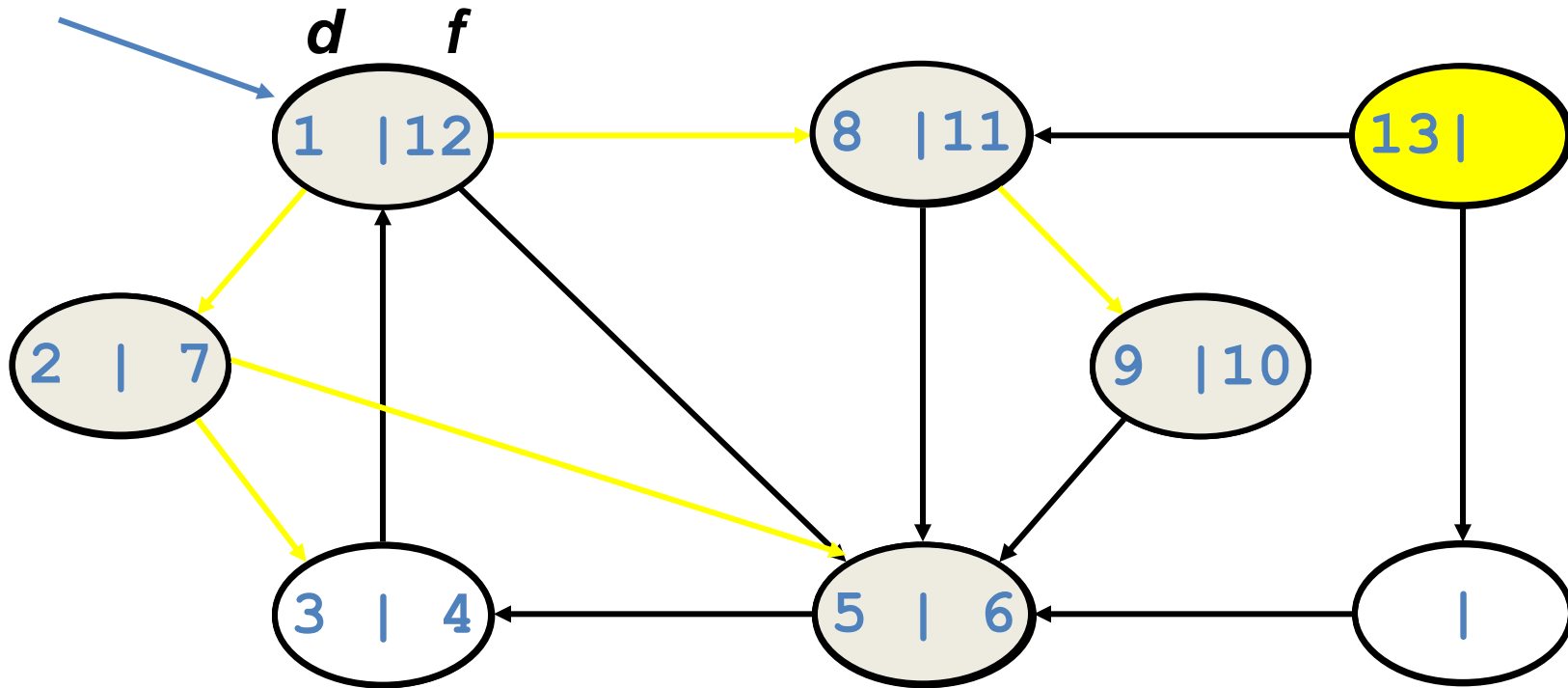
Παράδειγμα DFS

Πηγή



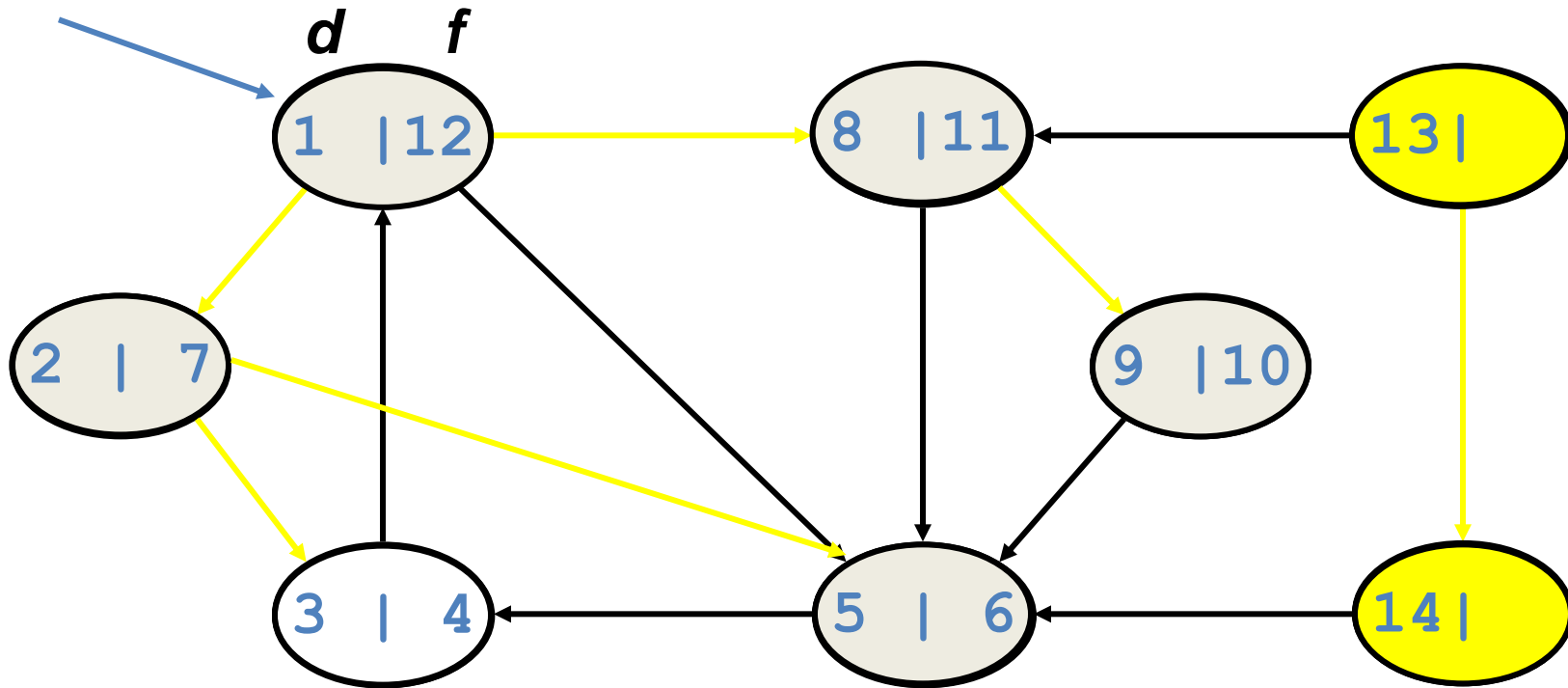
Παράδειγμα DFS

Πηγή



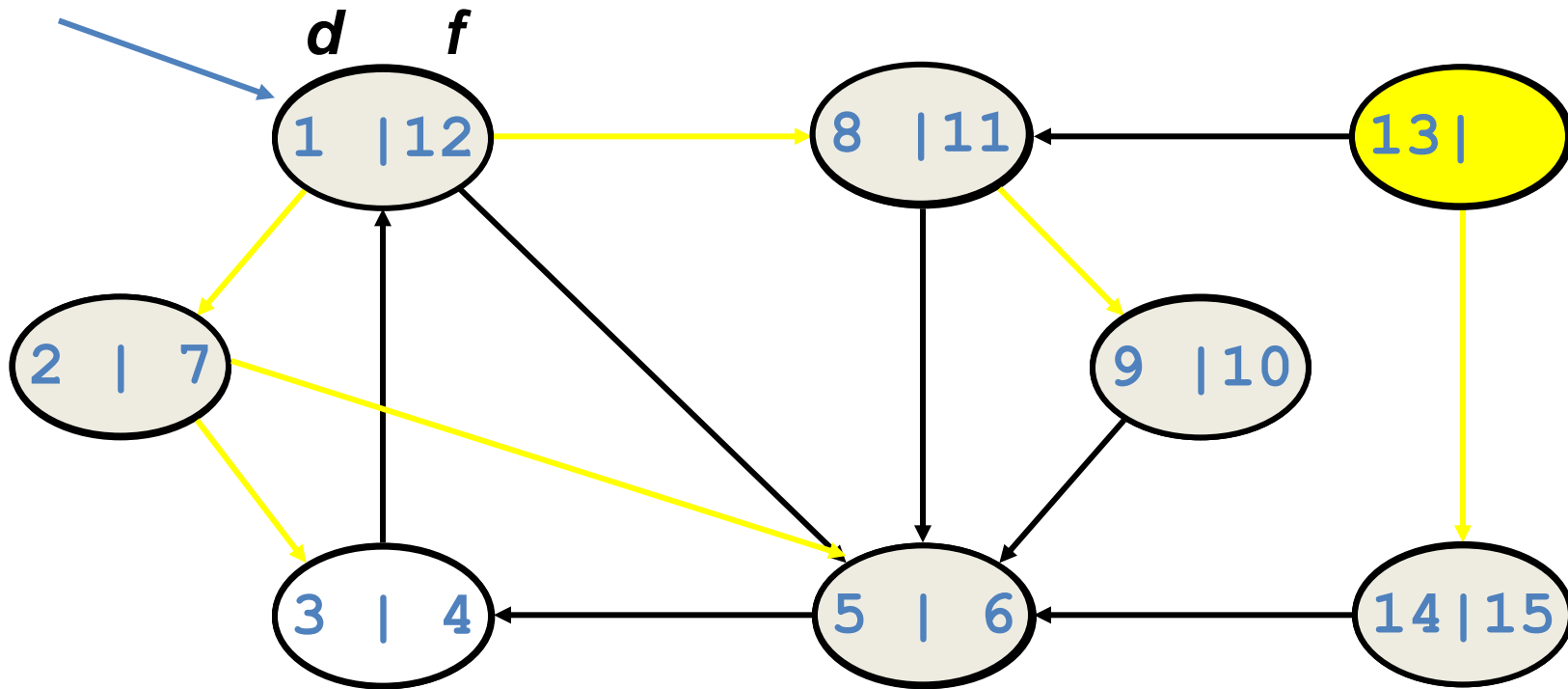
Παράδειγμα DFS

Πηγή



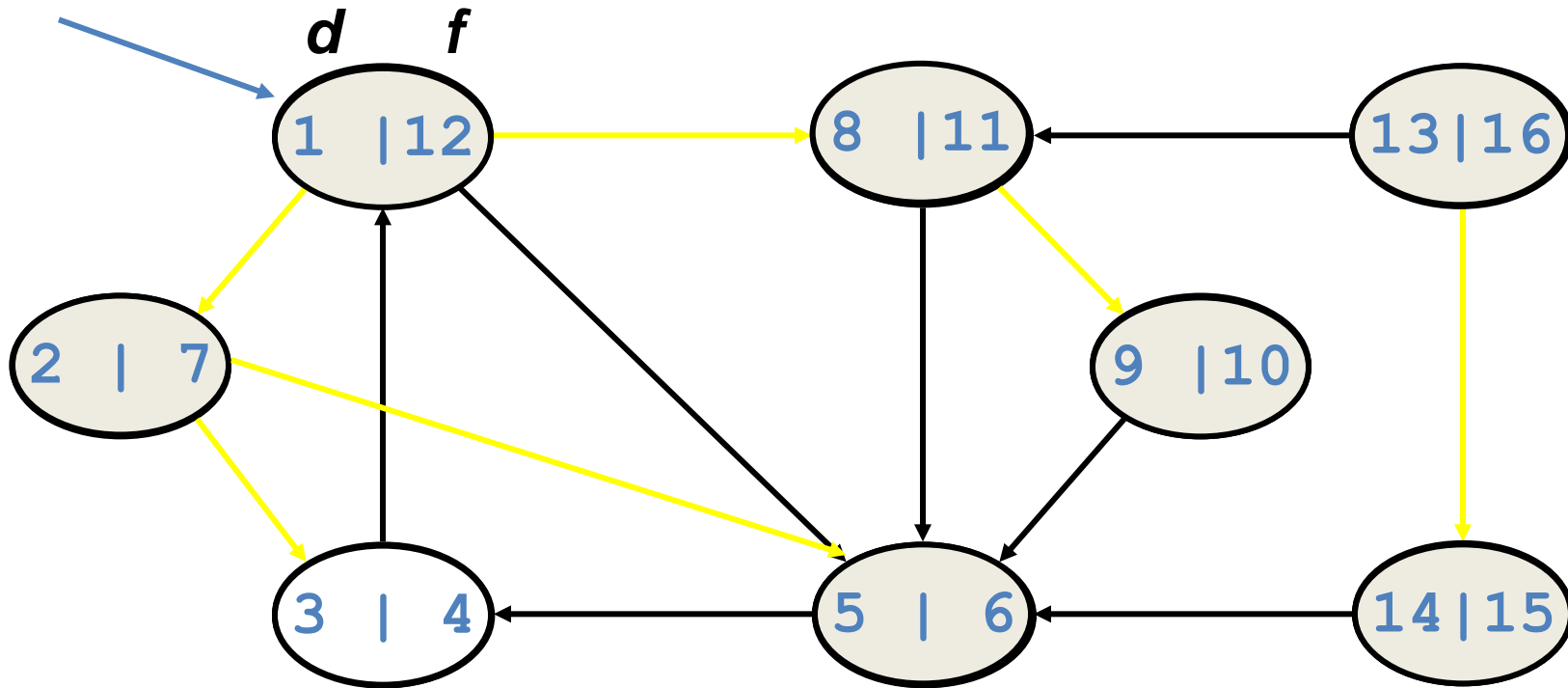
Παράδειγμα DFS

Πηγή



Παράδειγμα DFS

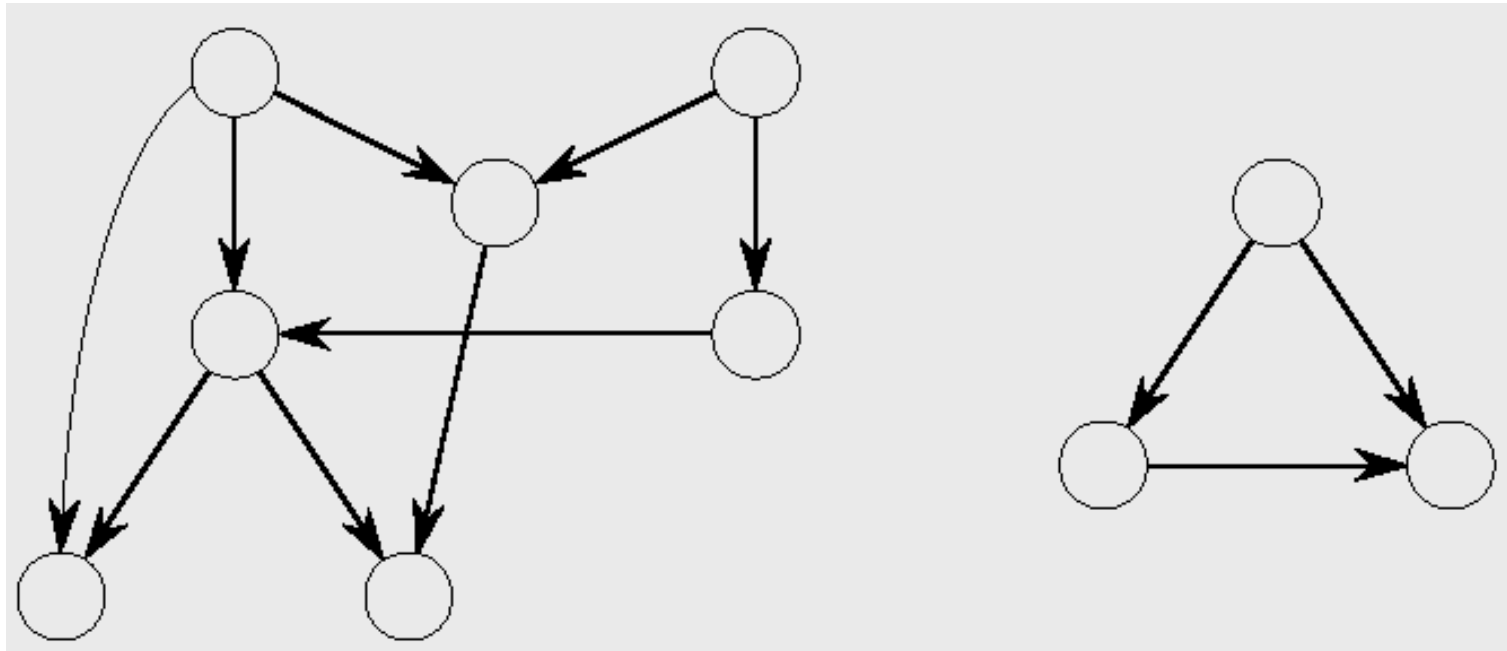
Πηγή



DAG

- DAG = Directed Acyclic Graph

Κατευθυνόμενος γράφος χωρίς κύκλους

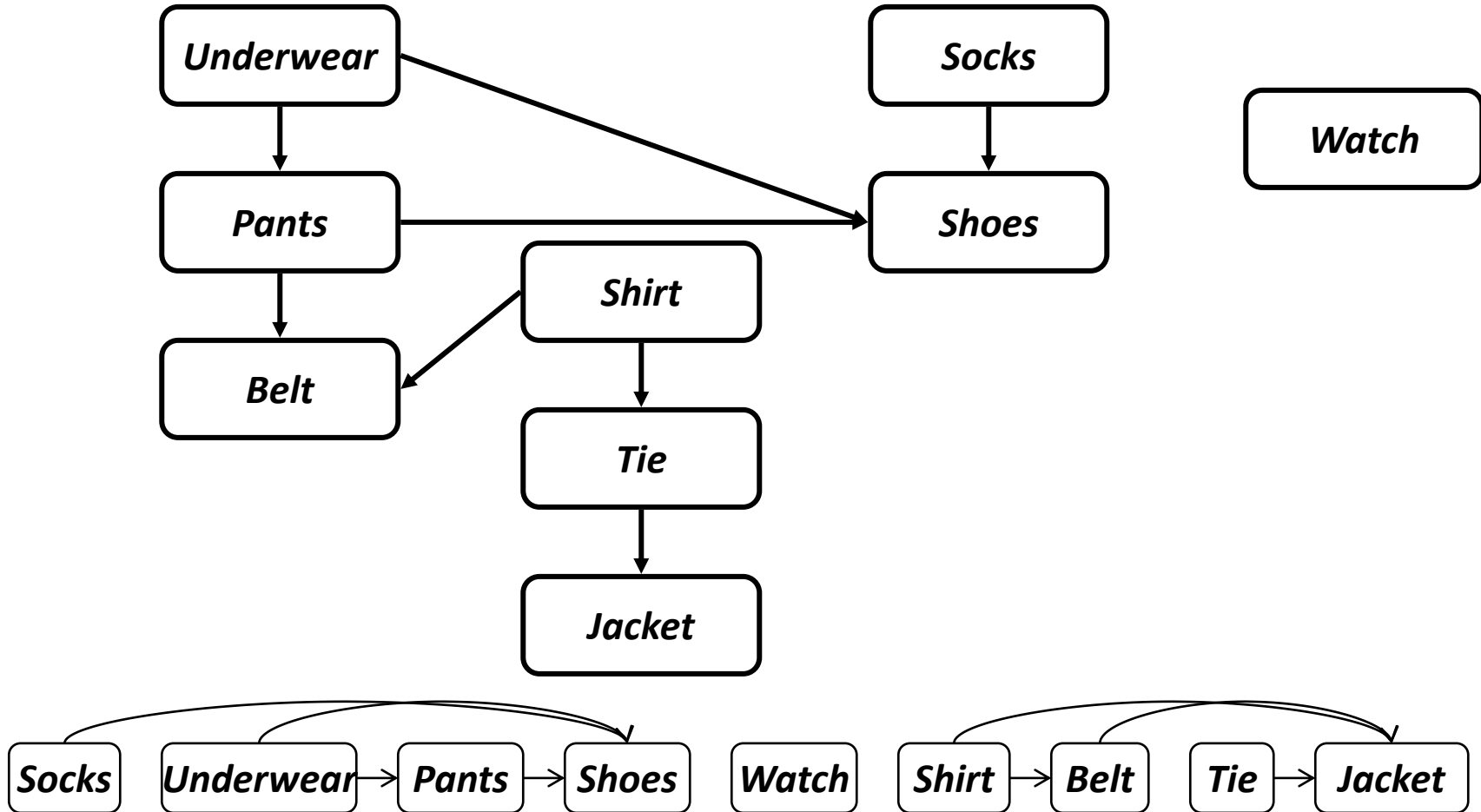


Βασικό Θεώρημα

- Ένας γράφος $G(V,E)$ είναι DAG εάν και μόνο εάν το δένδρο DFS που προκύπτει δεν έχει πίσω βέλη.



Τοπολογική Ταξινόμηση



Μέθοδος

- Topological-Sort()

{

Εκτέλεση DFS

Όταν ολοκληρώσουμε την επίσκεψη σε μία κορυφή, τη δίνουμε στην έξοδο.

Οι κορυφές δίδονται στην έξοδο με αντίστροφη σειρά.

}

- Κόστος: $O(V+E)$



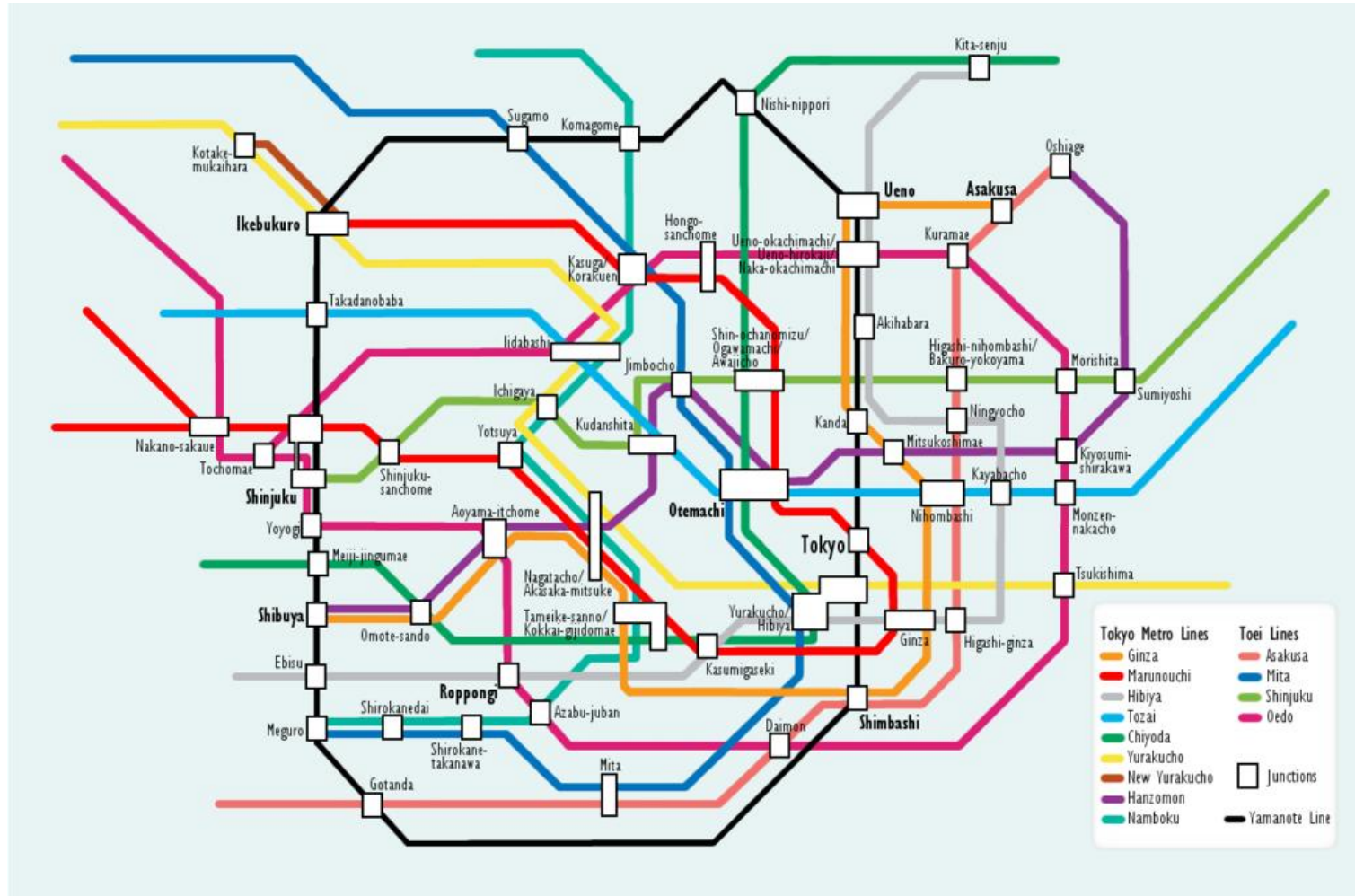
Ελάχιστα Μονοπάτια

- Πρόβλημα:

Σε ένα γράφο $G(V,E)$ με βάρη, δίνονται οι κορυφές A και B και ζητείται να προσδιορίσουμε ένα μονοπάτι που έχει το ελάχιστο κόστος μετάβασης από την A στη B .



Tokyo Subway Map



Ελάχιστα Ζευγνύοντα Δένδρα

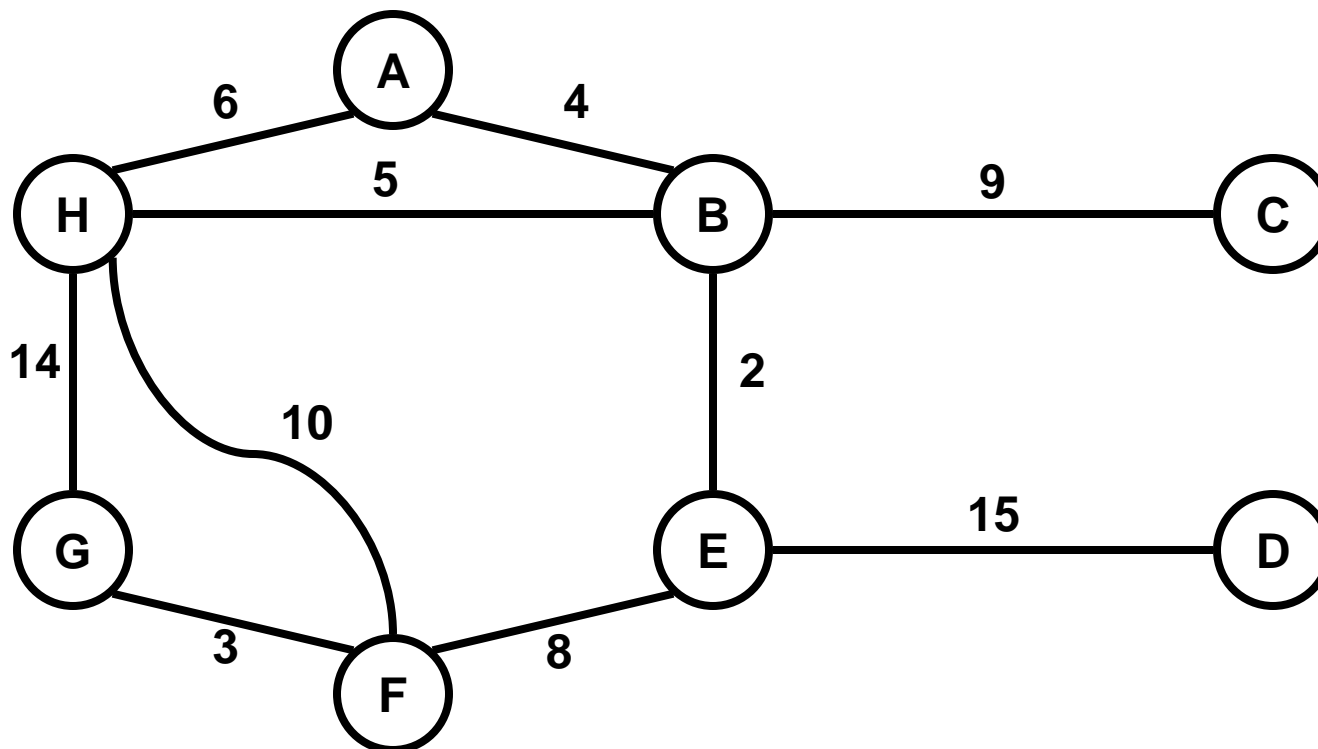
- Αλλιώς: Minimum Spanning Trees (MST)

Είναι το δένδρο που προκύπτει από έναν γράφο $G(V,E)$ αν κρατήσουμε τις ακμές που δίνουν αθροιστικά το ελάχιστο κόστος και ο γράφος που απομένει είναι συνδεδεμένος.

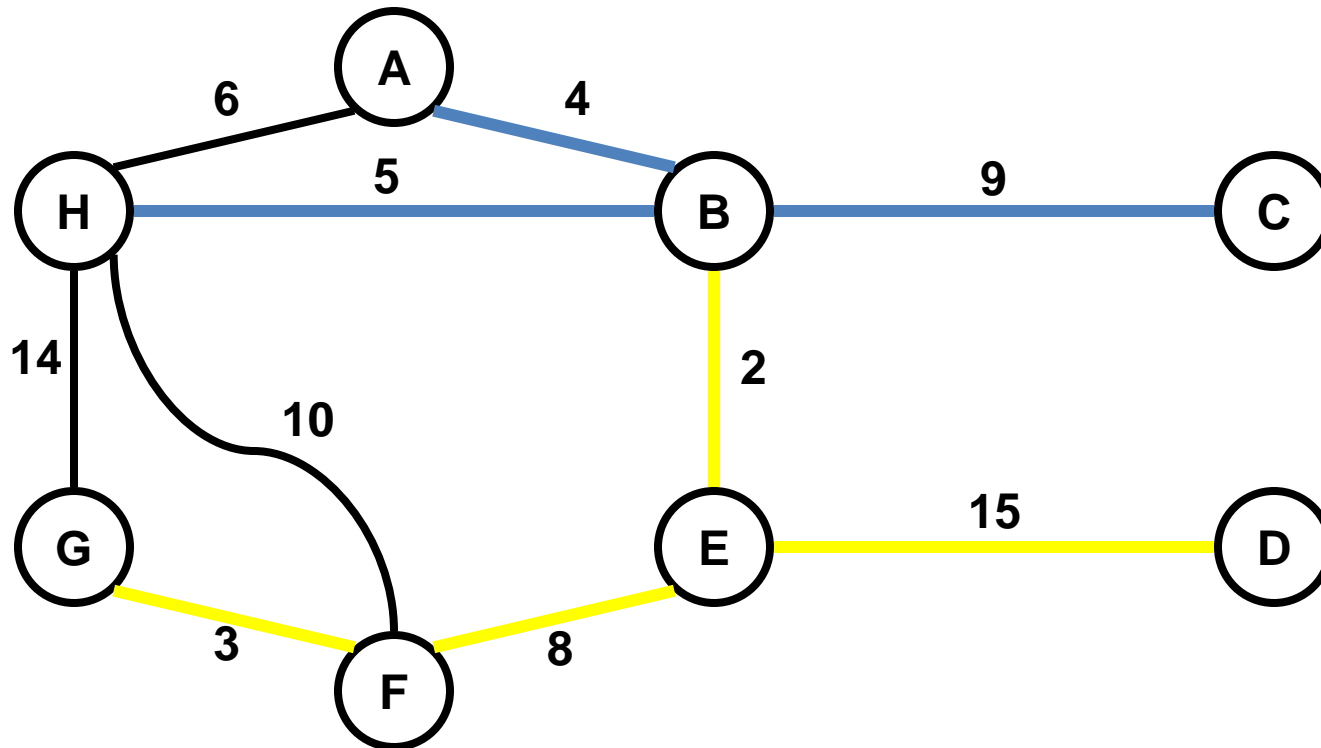


Minimum Spanning Tree

- Ποιες ακμές πρέπει να κρατήσουμε στο παρακάτω γράφημα για να έχουμε MST?



Minimum Spanning Tree



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

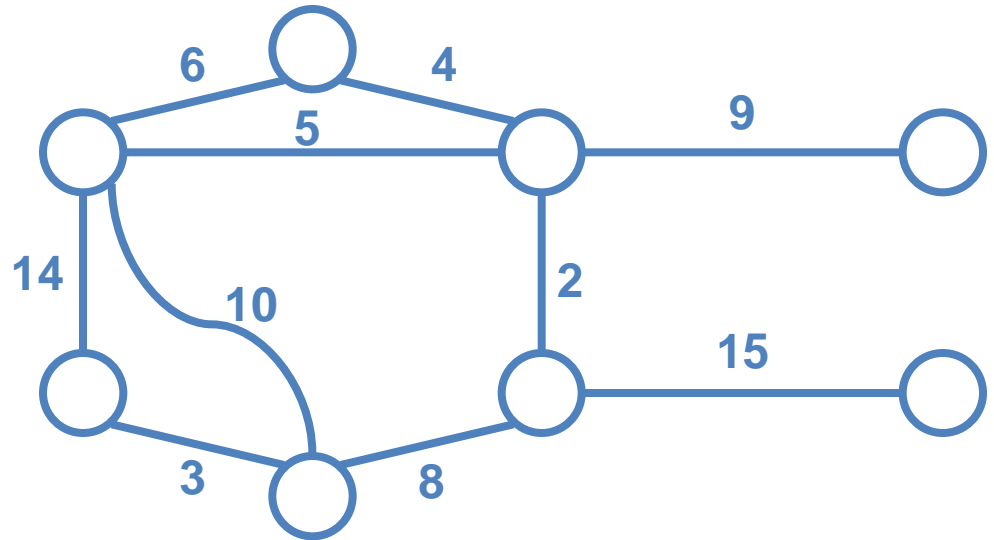
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Run on example graph



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

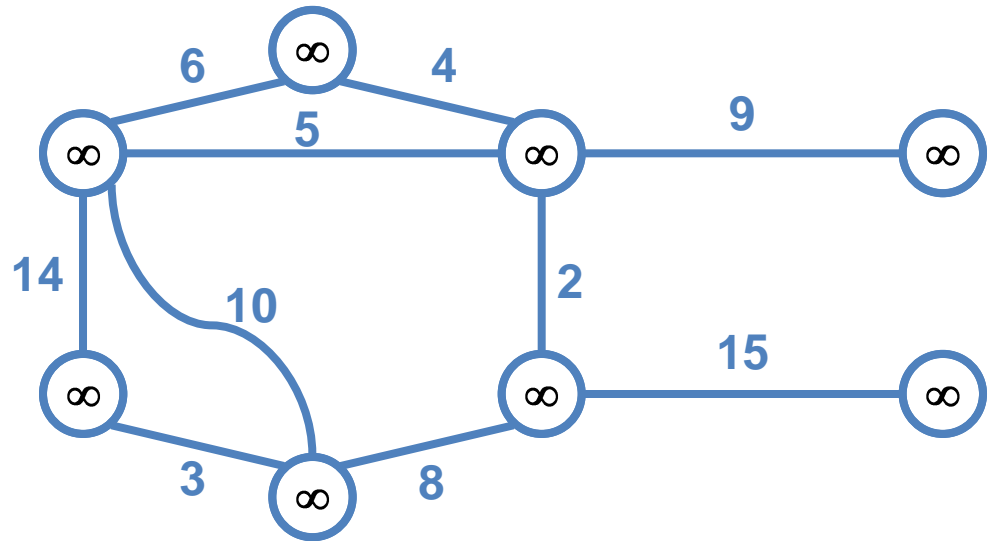
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Run on example graph



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

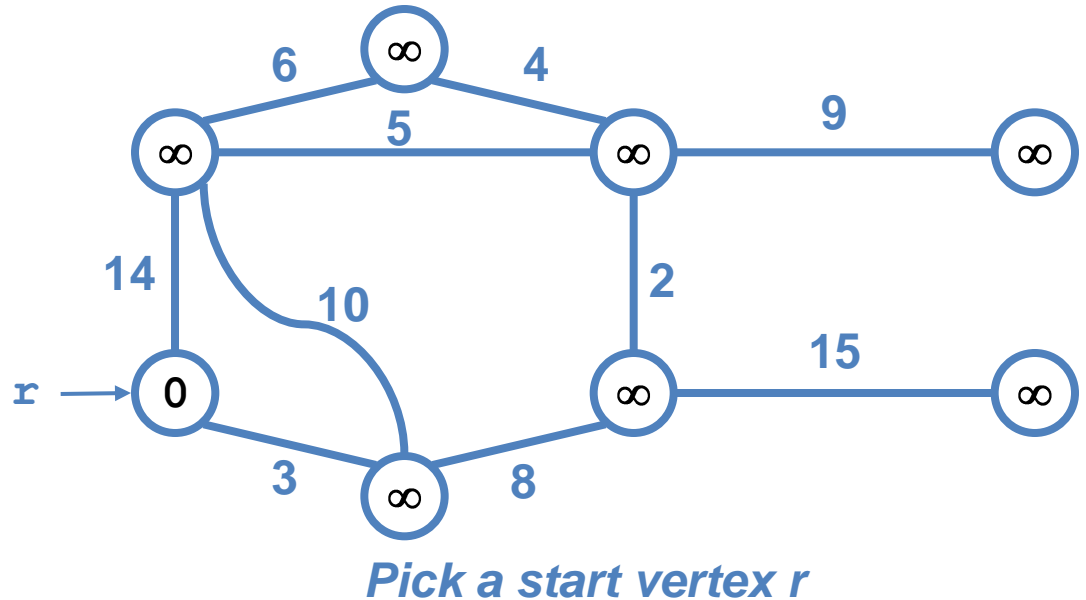
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

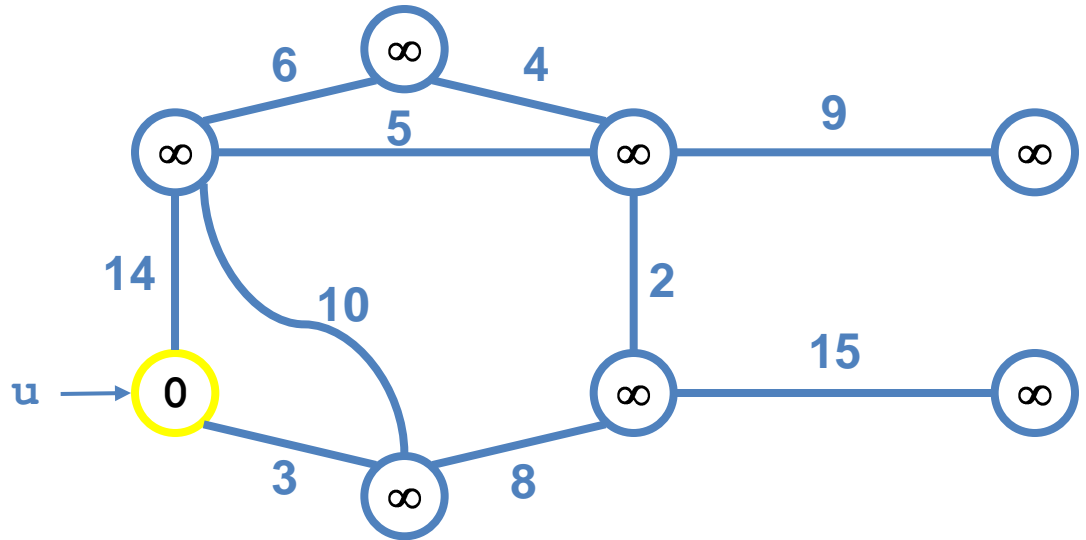
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

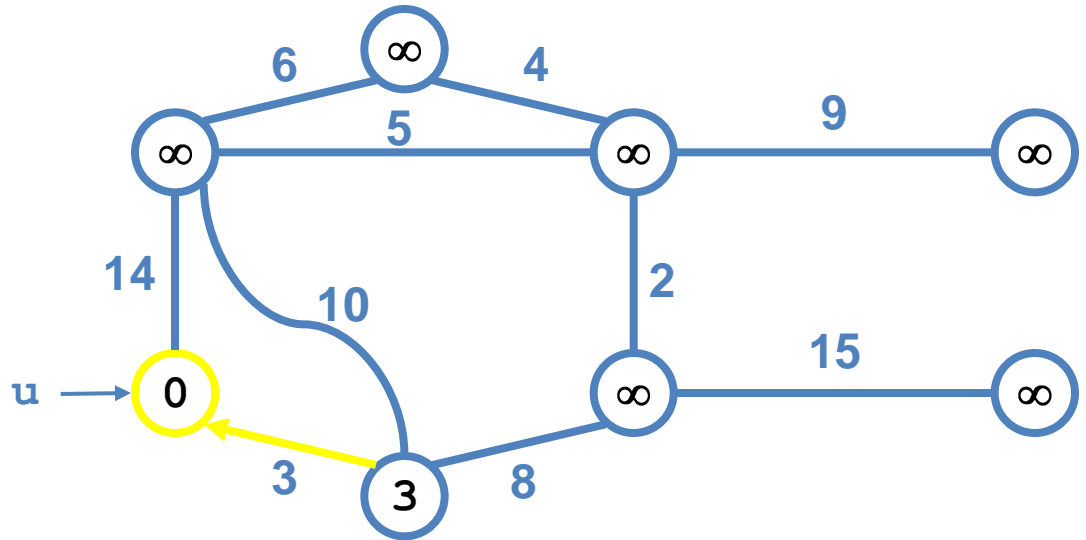
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

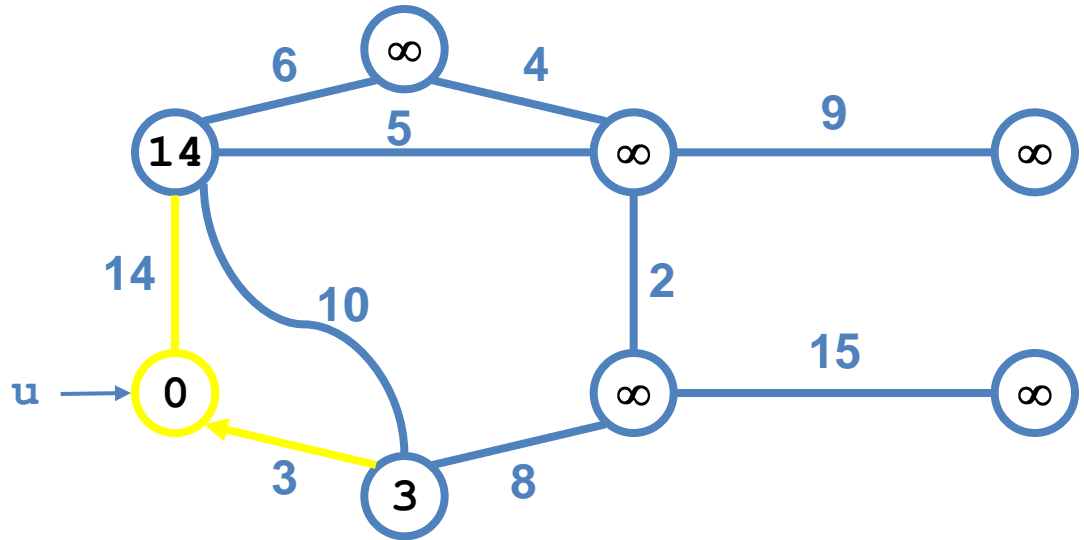
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

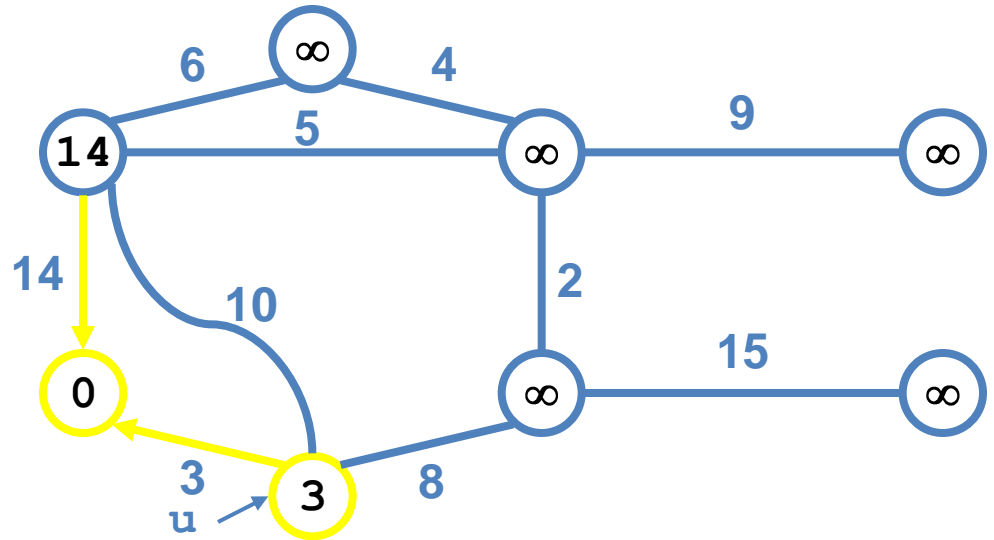
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

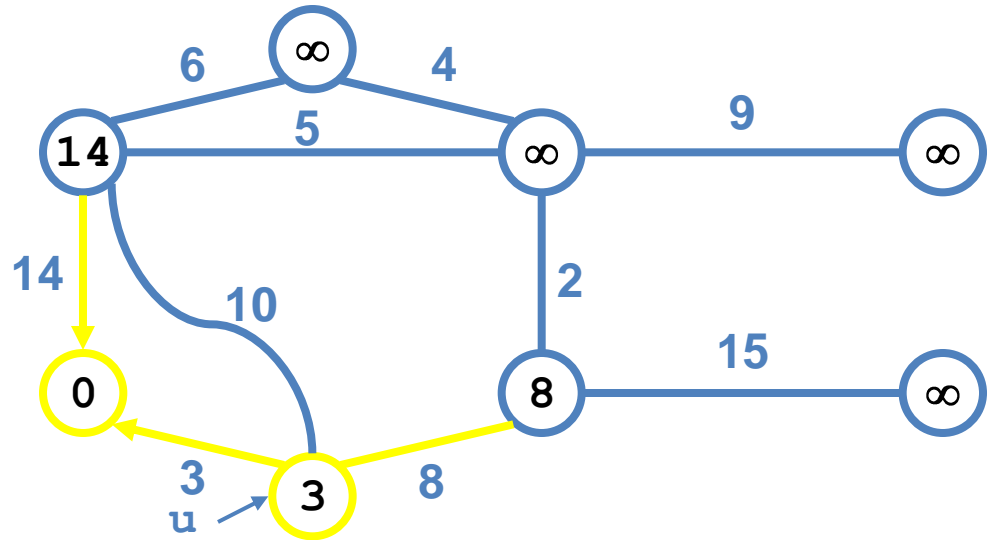
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

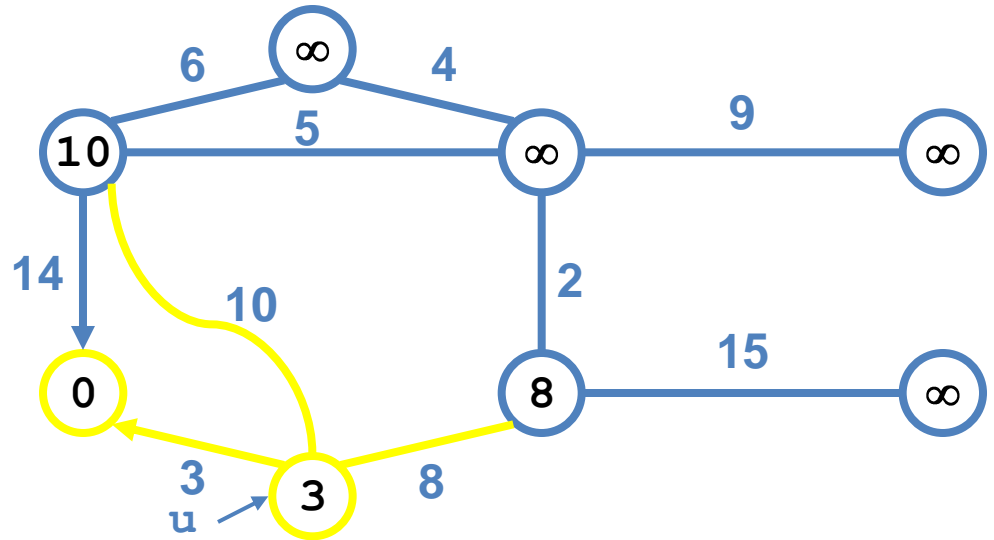
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

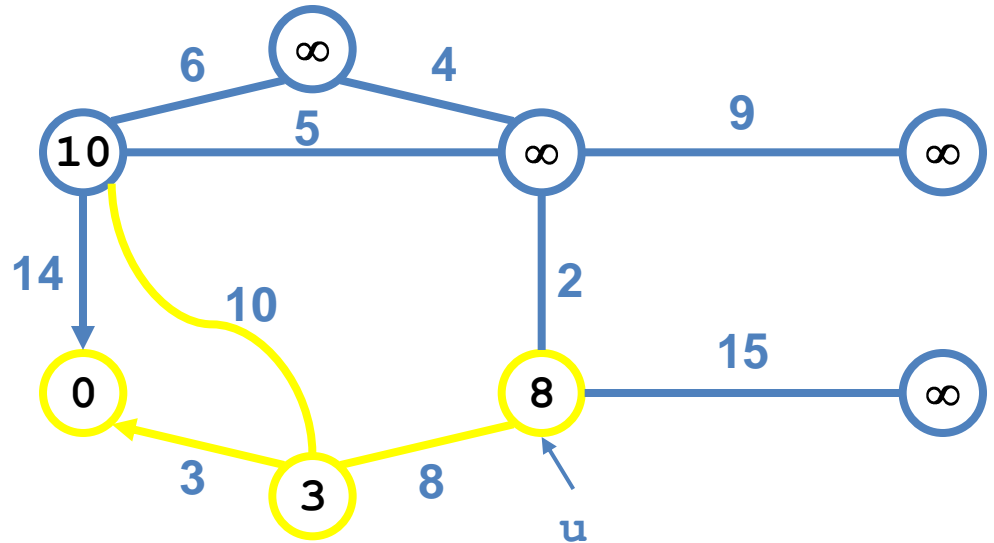
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

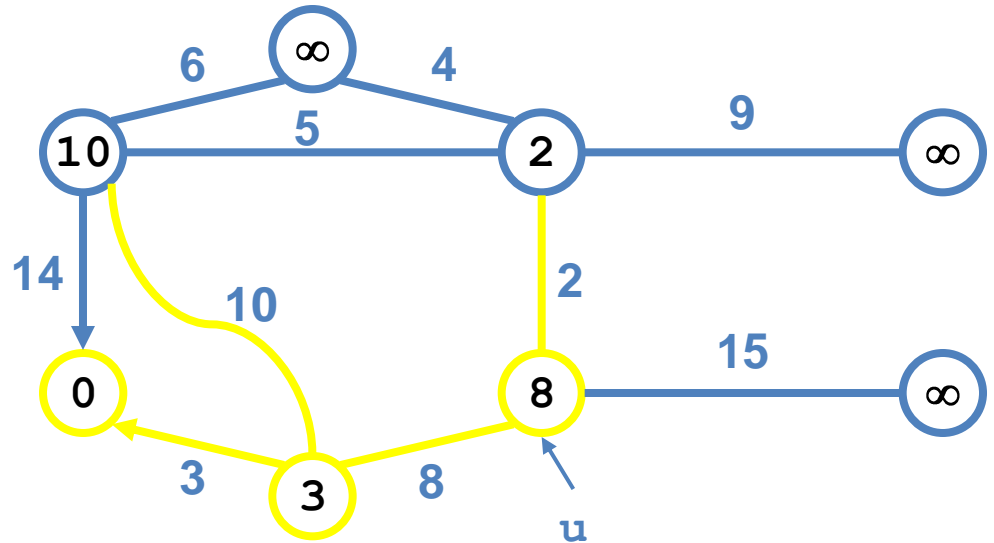
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

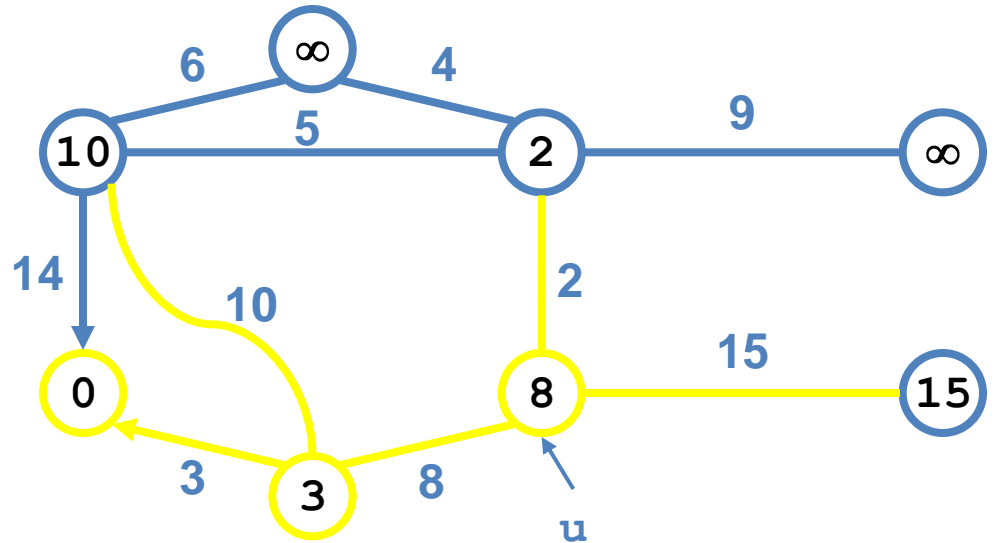
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

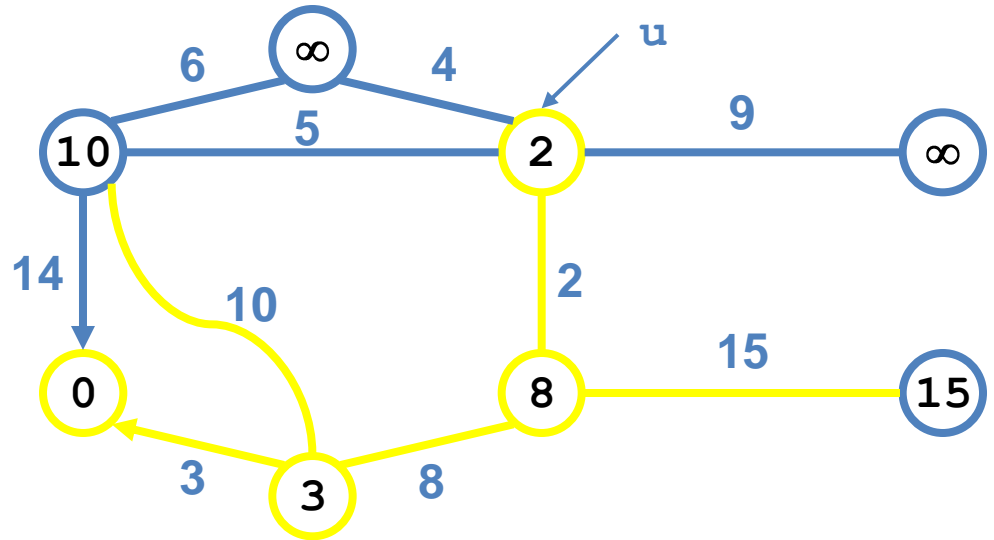
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

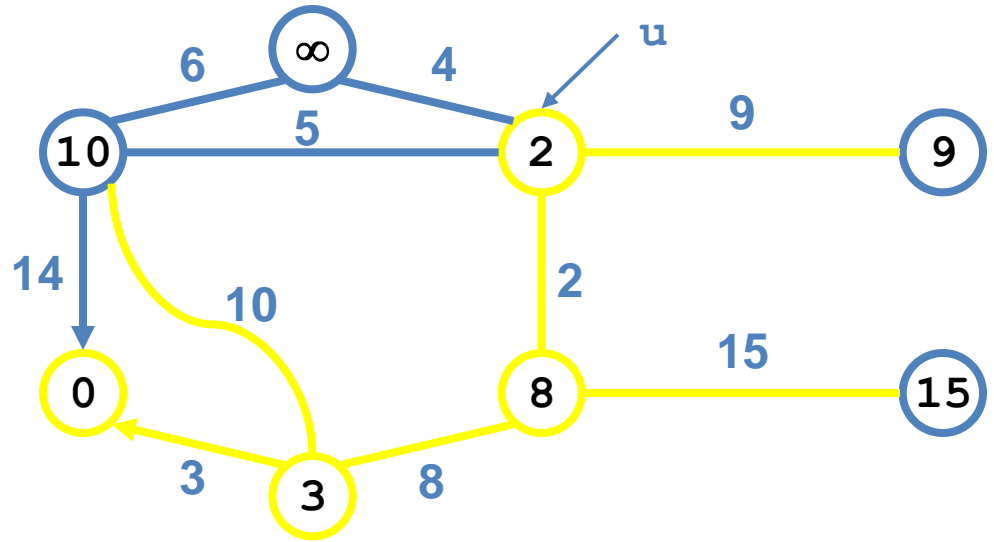
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

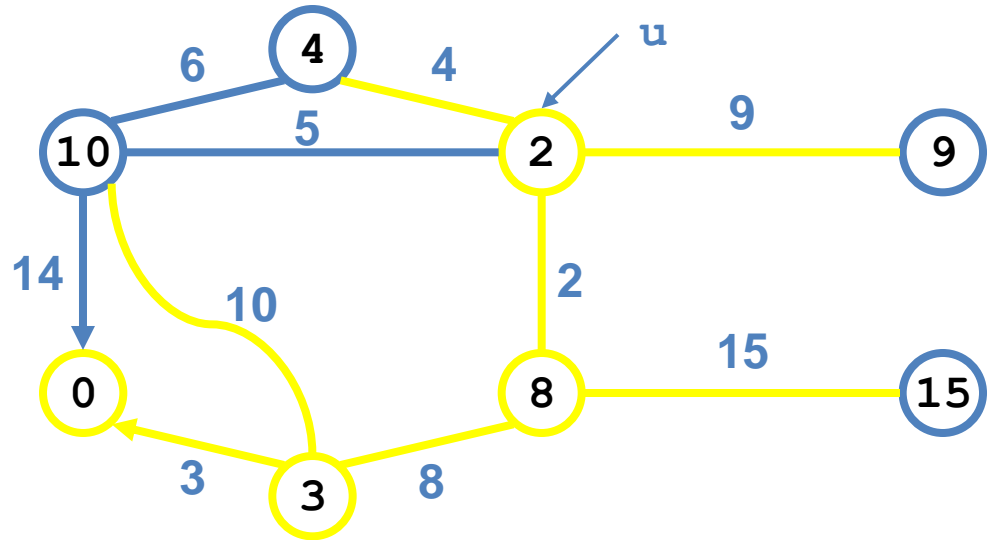
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

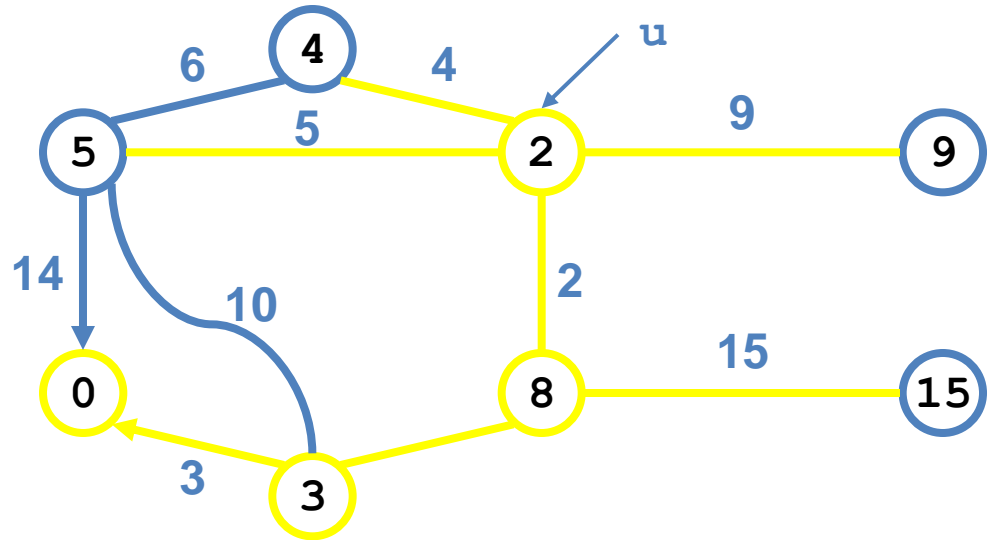
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

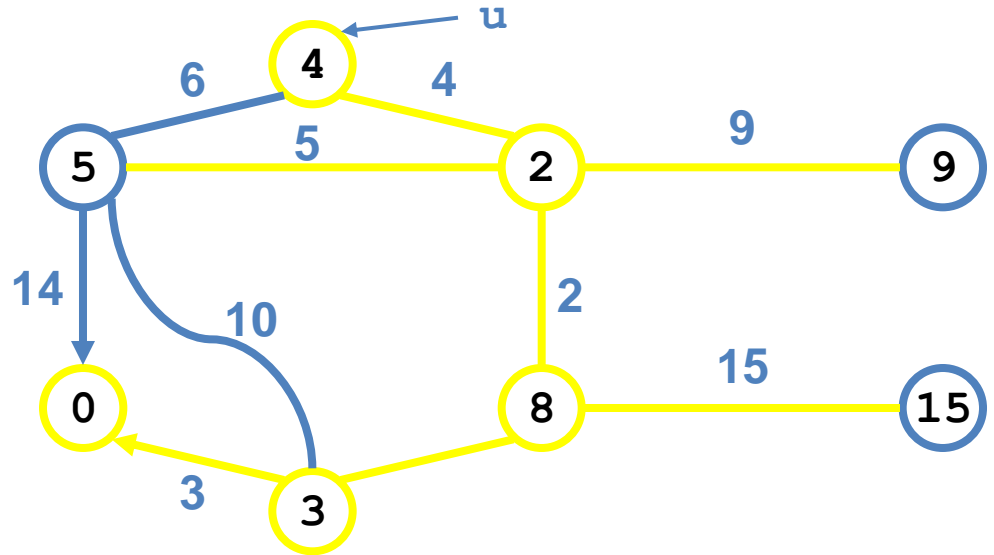
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

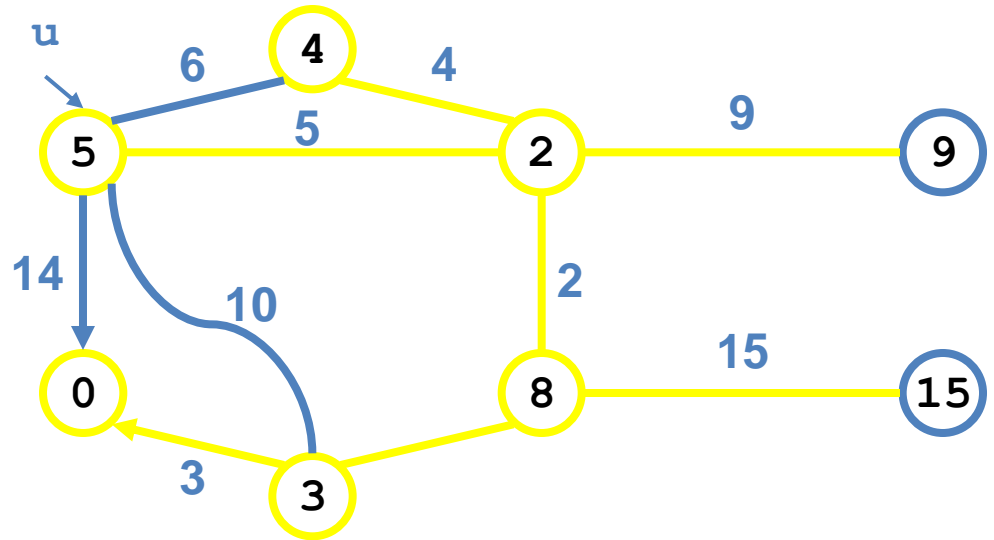
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

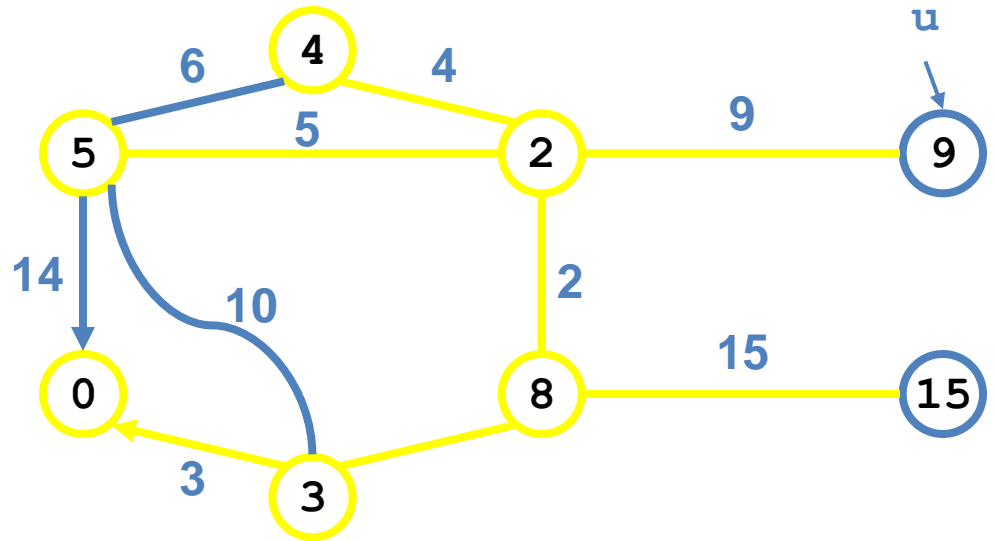
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

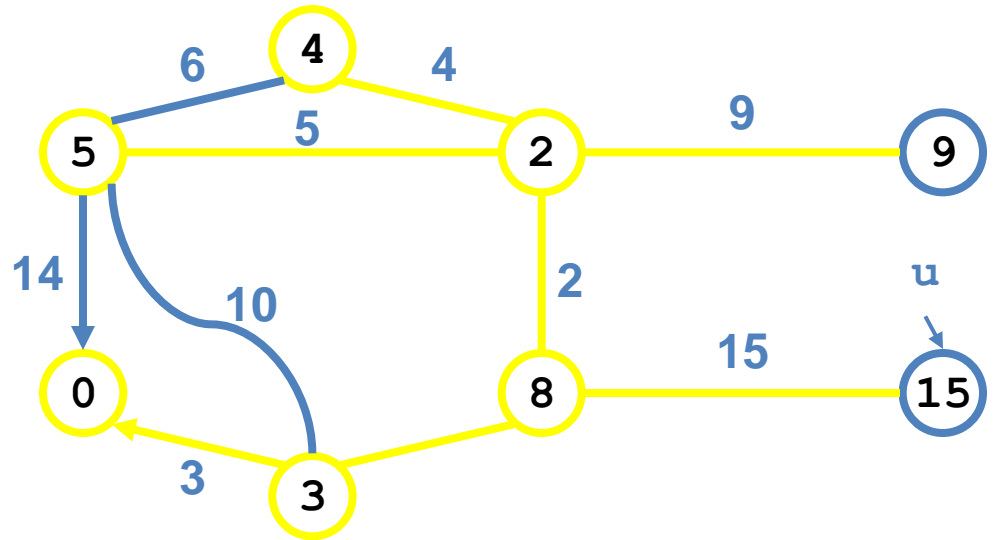
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < key[v]$)

$p[v] = u;$

$key[v] = w(u,v);$



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Απόστολος Παπαδόπουλος. «Δομές Δεδομένων. Γράφοι». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:
<http://eclass.auth.gr/courses/OCRS389/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 4.0 [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.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>





Τέλος ενότητας

Επεξεργασία: <Μαυρίδης Απόστολος>
Θεσσαλονίκη, <Εαρινό εξάμηνο 2013-2014>



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Σημειώματα

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.

