Αναζήτηση κατά βάθος (Depth-First Search DFS)

- Η DFS εξερευνά 'πιο βαθιά' το γράφημα, όταν αυτό είναι δυνατό.

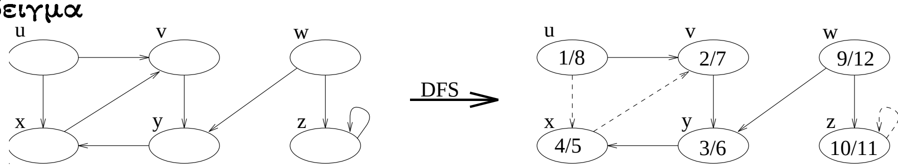
- Εξερευνούνται οι ακμές που ξεκινούν απο τον πιο πρόσφατα ανακαλυφθέντα κόμβο u ο οποίος έχει ακόμη ανεξευρεύνητες ακμές.

- Η DFS σφραγίζει χρονικά κάθε κόμβο.

  $d[u]$: η χρονική στιγμή οπου ο u ανακαλύπτεται για πρώτη φορά.
  $f[u]$: η χρονική στιγμή οπου η εξερεύνηση μέσω του u ολοκληρώνεται, δηλ., η λίστα των γειτονικών με τον u κόμβων έχει πλήρως εξερευνηθεί.

- Η DFS παράγει ένα δάσος από δένδρα (*depth-first forest*) το οποίο αποτελείται από ένα η περισσότερα 'δένδρα κατά-βάθος αναζήτησης' (*depth-first trees*).

Παράδειγμα



- Κατά την εκτέλεσή της, η DFS χρωματίζει τους κόμβους *άσπρους*, *γκρι*, ή *μαύρους*.

  Άσπροι κόμβοι: δεν έχουν ανακαλυφθεί έως τώρα.

  Γκρι κόμβοι: έχουν ανακαλυφθεί, αλλά υπάρχουν ακόμη ανεξερεύνητες ακμές που εξέρχονται από ατούς.

  Μαύροι κόμβοι: έχουν ανακαλυφθεί, και όλες οι εξερχόμενες από αυτούς ακμές έχουν εξερευνηθεί.

DFS(G)

for each vertex $u \in V[G]$
  do $color[u] = white$
    $\pi[u] = \mathrm{NI\Lambda}$
$time = 0$
for each vertex $u \in V[G]$
  do if $color[u] = white$
      then DFS_visit(u)

DFS_visit(u)

$color[u] = grey$

$time = time + 1$

$d[u] = time$

for each vertex $v \in Adj[u]$

  do if $color[v] = white$

      then $\pi[v] = u$

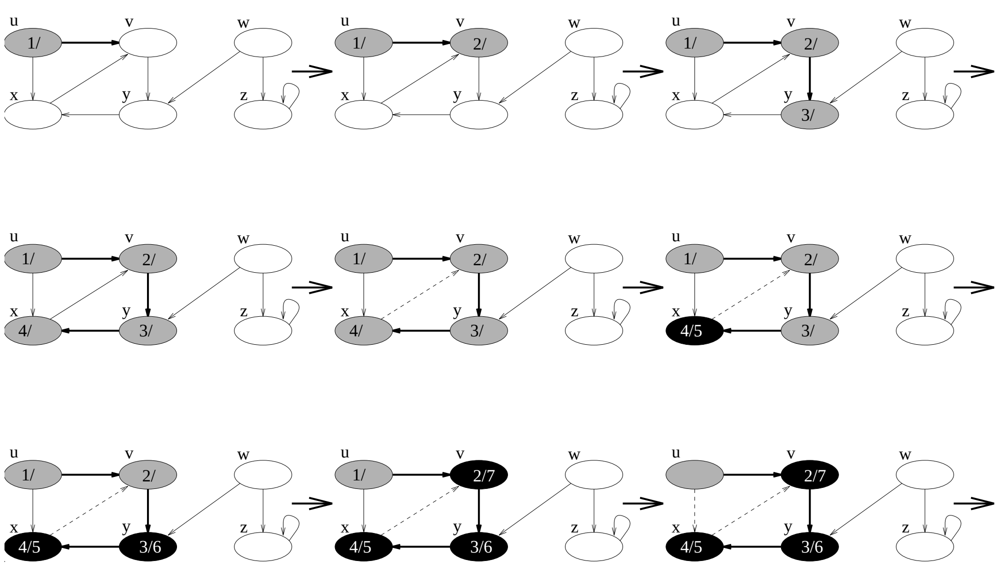
        DFS_visit(v)

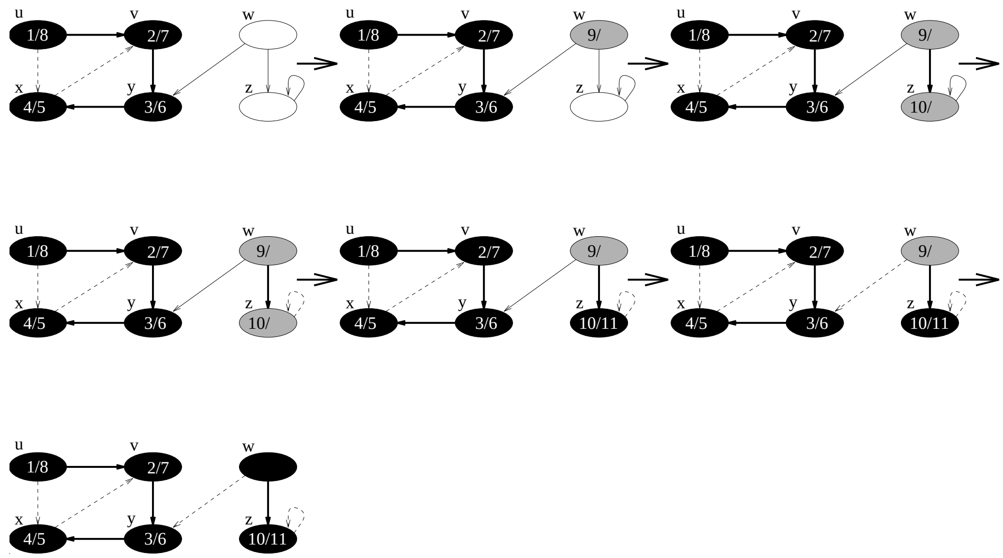
$color[u] = black$

$time = time + 1$

$f[u] = time$

Ανάλυση: $O(V + E)$





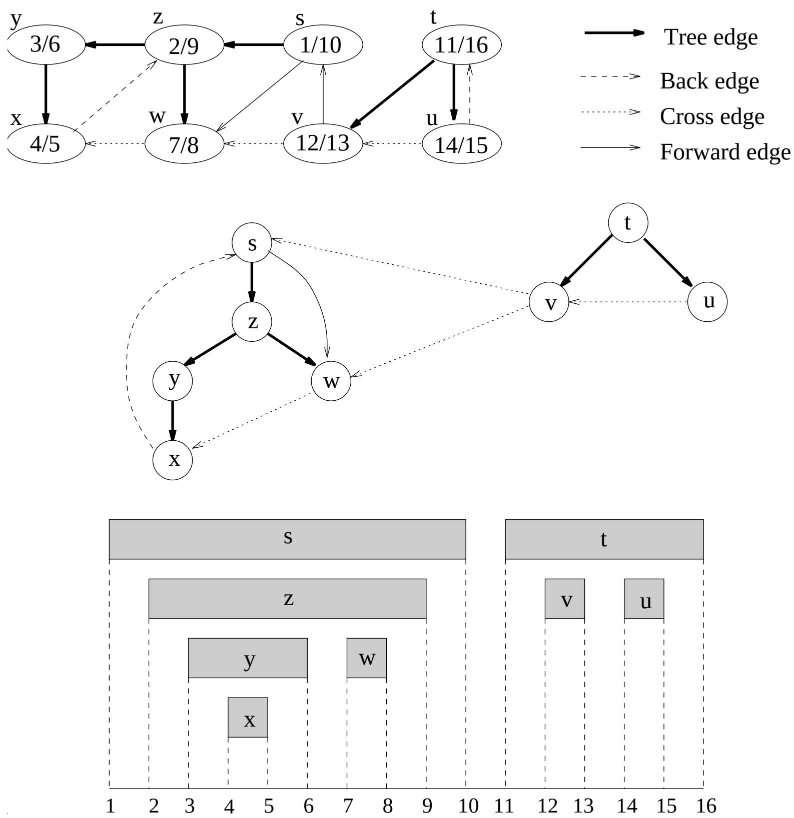
Ιδιότητες της DFS

- **Κατηγορίες ακμών**

  1. *Ακμές δένδρου (tree edges)* είναι οι ακμές που ανήκουν στο κατά-βάθος δάσος δένδρων G_π. Η ακμή (u, v) είναι μία ακμή δένδρου εάν ο κόμβος v έχει ανακαλυφθεί κατά την εξερεύνηση της ακμής (u, v).
  2. *Ακμές προς τα πίσω (back edges)* είναι οι ακμές (u, v) οι οποίες ενώνουν ένα κόμβο u με ένα πρόγονο του v σε κάποιο κατά-βάθος δένδρο αναζήτησης.
  3. *Ακμές προς τα εμπρός (forward edges)* είναι οι ακμές (u, v) οι οποίες ενώνουν ένα κόμβο u με έναν απόγονό του v σε κάποιο κατά-βάθος δένδρο αναζήτησης.
  4. *Ακμές προς τα πλάγια (cross edges)* είναι όλες οι άλλες ακμές. Είναι ακμές μεταξύ των κόμβων του ίδιου δένδρου, υπό την προϋπόθεση ότι ο ένας κόμβος δεν είναι πρόγονος του άλλου, ή είναι ακμές μεταξύ κόμβων που ανήκουν σε διαφορετικά δένδρα κατά-βάθους αναζήτησης.

- **Παρενθετική δομή**

  Οι χρονικές στιγμές *ανακάλυψης* και *ολοκλήρωσης* σχηματίζουν μια δομή παρενθέσεων.



Πως κατηγοριοποιούνται οι ακμές

- Κάθε ακμή (u, v) μπορεί να κατηγοριοποιηθεί με βάση το χρώμα που έχει ο κόμβος v κατά την εξερεύνησή της.

$$(u,v) \text{ είναι } \begin{cases} \textit{Ακμή δένδρου} & \text{εάν ο } v \text{ είναι άσπρος.} \\ \textit{Ακμή προς τα πίσω} & \text{εάν ο } v \text{ είναι γκρι.} \\ \textit{Ακμή προς τα εμπρός} & \text{εάν ο } v \text{ είναι μαύρος και } d[u] < d[v]. \\ \textit{Ακμή προς τα πλάγια} & \text{εάν ο } v \text{ είναι μαύρος και } d[u] > d[v]. \end{cases}$$

Θεώρημα Κατά την κατά-βάθος αναζήτηση ενός μη-κατευθυνόμενου γραφήματος G κάθε ακμή του G κατατάσσεται σαν ακμή δένδρου ή ακμή προς τα πίσω.

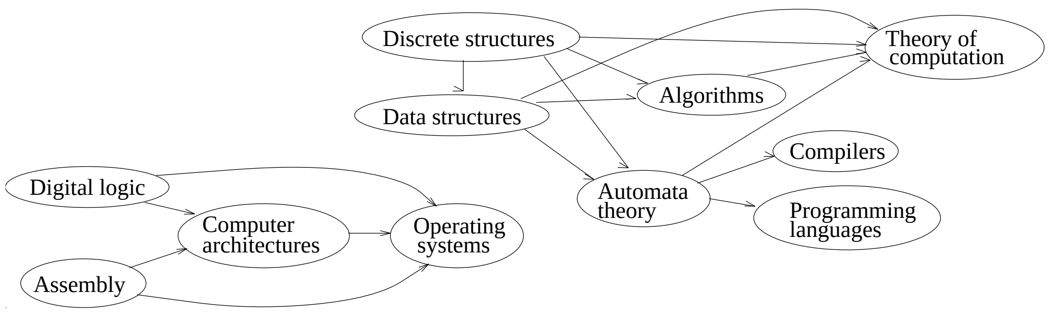
Τοπολογική διάταξη (Topological sort)

- Η τοπολογική διάταξη ενός ακυκλικού κατευθυνόμενου γραφήματος (DAG) $G = (V, E)$ είναι μια γραμμική διάταξη όλων των κόμβων του τέτοια ώστε εάν ο G περιέχει την ακμή (u, v) τότε ο κόμβος u εμφανίζεται πριν από τον κόμβο v.

- Εάν το γράφημα δεν είναι ακυκλικό, τότε δεν είναι δυνατή η τοπολογική του διάταξη.

Εφαρμογή Μοντελοποιεί την σειρά εκτέλεσης γεγονότων.

Παράδειγμα Προαπαιτούμενα μαθήματα.

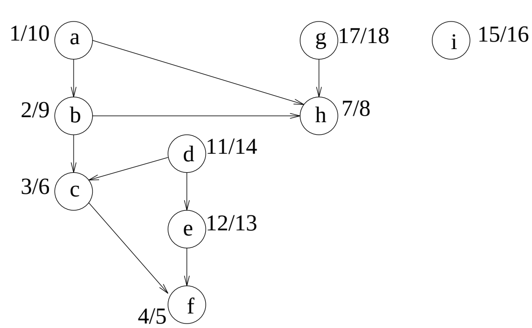


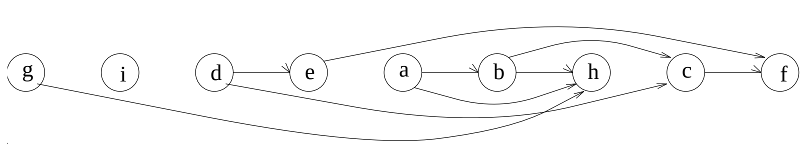
Topological_Sort(G)

1. Εκτέλεσε την DFS(G) και υπολόγισε τη χρονική στιγμή ολοκλήρωσης $f[u]$ για κάθε κόμβο u.
2. Οταν ολοκληρώνεται η αναζήτηση μέσω ενός κόμβου, εισήγαγε τον κόμβο στην ‘αρχή’ μιας συνδεδεμένης λίστας.
3. Επέστρεψε την συνδεδεμένη λίστα.

Ανάλυση $O(V + E)$

Παράδειγμα

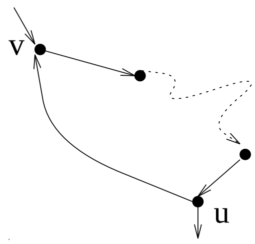




Λήμμα Ένα κατευθυνόμενο γράφημα G είναι ακυκλικό εάν και μόνο εάν η κατά βάθος αναζήτηση του G δεν παράγει 'προς τα πίσω' ακμές.

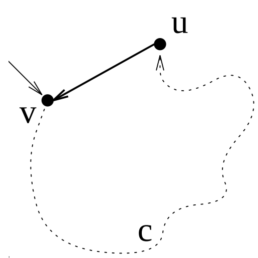
Απόδειξη

'$\Longrightarrow$' Το G είναι ακυκλικό $\Rightarrow$ Το G δεν έχει 'προς τα πίσω' ακμές.



Υπέθεσε ότι υπάρχει μια 'προς τα πίσω' ακμή (u, v). Τότε, ο v είναι πρόγονος του u στο κατά-βάθος δάσος αναζήτησης. Άρα, υπάρχει ένα μονοπάτι από τον v προς τον u στο G. Το μονοπάτι αυτό μαζί με την (u, v) σχηματίζει ένα κύκλο. $\rightarrow\leftarrow$

'$\Longleftarrow$' Το G δεν έχει 'προς τα πίσω' ακμές $\Rightarrow$ Το G είναι ακυκλικό



Υπέθεσε ότι το G περιέχει ένα κύκλο c. Θα δείξουμε ότι η κατά-βάθος αναζήτηση παράγει μια 'προς τα πίσω' ακμή. Έστω v ο πρώτος κόμβος που ανακαλύπτεται και ανήκει στο κύκλο c. Με μία κατάλληλη αναδιάταξη των κόμβων του c στις λίστες γειτονικών κόμβων, μπορούμε να κάνουμε την DFS να επισκεφθεί τον u και έτσι ο u γίνεται απόγονος του v. Τότε, η (u, v) γίνεται μια 'προς τα πίσω' ακμή. $\rightarrow\leftarrow$

Θεώρημα Η Topological_Sort(G) παράγει μία τοπολογική διάταξη ενός DAG G

Απόδειξη

Αρκεί να δείξουμε ότι για κάθε ζευγάρι κόμβων $u,\ v \in V$, εάν υπάρχει μία ακμή στο G από τον u προς τον v , τότε $f[v] < f[u]$.

Εξετάζουμε την οποιαδήποτε ακμή (u, v) η οποία εξερευνείται από την DFS(G).

Όταν εξερευνείται η (u, v) , **ο v δεν μπορεί να είναι γκρι.**
(ο v θα ήταν πρόγονος του u, και έτσι, η (u, v) θα ήταν μία 'προς τα πίσω' ακμή).

Άρα, ο v είναι άσπρος ή μαύρος.

- **Εάν ο v είναι άσπρος**, τότε γίνεται απόγονος του u. $\Longrightarrow f[v] < f[u]$.
- **Εάν ο v είναι μαύρος**, τότε η αναζήτηση μέσω του v έχει ολοκληρωθεί. $\Longrightarrow f[v] < f[u]$.

$\square$