Αλγόριθμοι και Πολυπλοκότητα

7ο εξάμηνο

Σ.Η.Μ.Μ.Υ. & Σ.Ε.Μ.Φ.Ε.

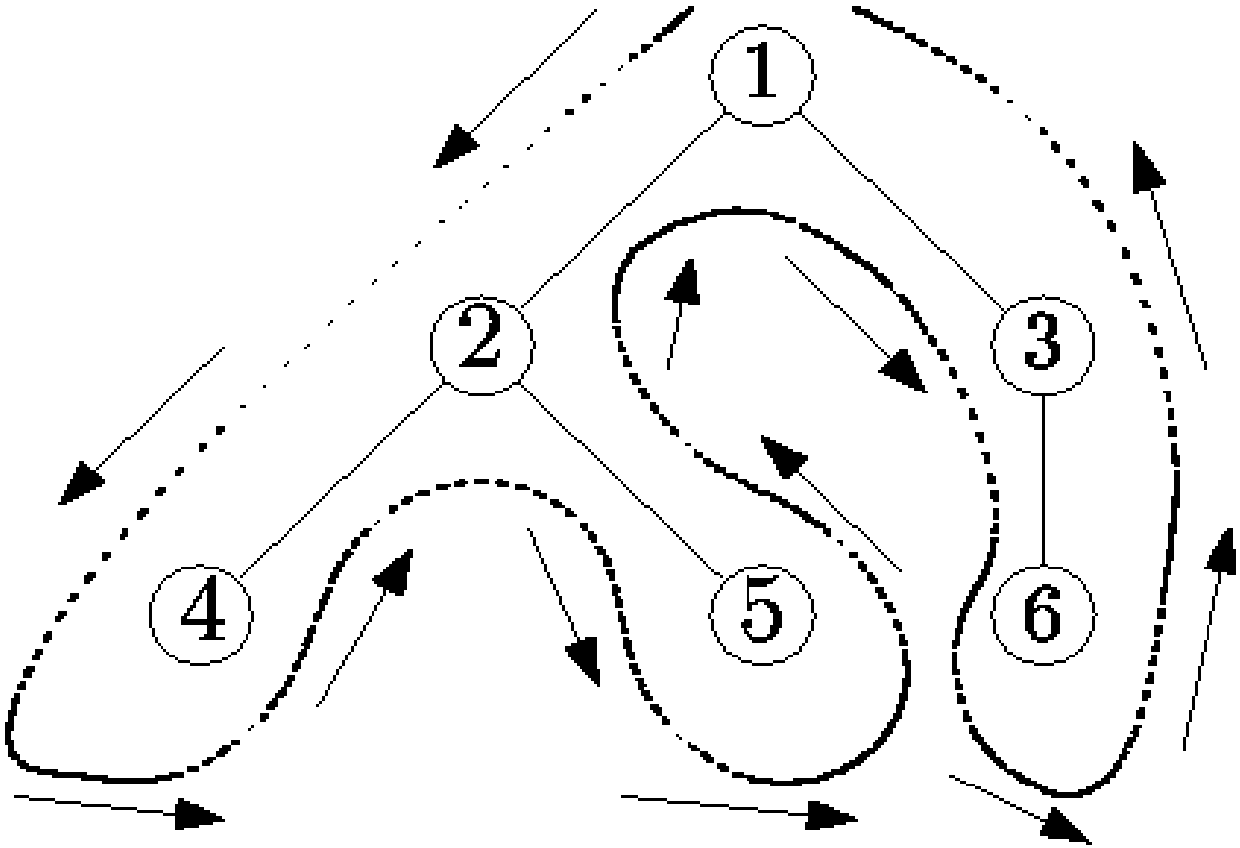
http://www.corelab.ece.ntua.gr/courses/

6η εβδομάδα: Διάσχιση και Αναζήτηση σε Δένδρα και Γράφους (BFS, DFS)

Διδάσκοντες:

Στάθης Ζάχος – Άρης Παγουρτζής

Διάσχιση Δένδρου



Διάσχιση Δένδρου (συν.)

- προδιατεταγμένη: 1 2 4 5 3 6 (preorder)
- μεταδιατεταγμένη: 4 5 2 6 3 1 (postorder)
- ενδοδιατεταγμένη: 4 2 5 1 6 3 (inorder)

Ενδοδιατεταγμένη διάσχιση

```
procedure Inorder (T: bintree);
begin
   if T<>empty then
   begin
      Inorder(leftchlild(T)); Visit(T);
      Inorder(rightchild(T))
   end
end
```

Προδιατεταγμένη διάσχιση

```
procedure Preorder (T:bintree);
begin
   if T<>empty then
   begin
      Visit(T); Preorder(leftchlild(T));
      Preorder(rightchild(T))
   end
end
```

Μεταδιατεταγμένη διάσχιση

```
procedure Postorder (T);
begin
  while T<>empty do
  begin
    Postorder(leftchild(T)); Postorder(rightchild(T);
    Visit(T)
  end
end
```

Ενδοδιατεταγμένη διάσχιση με 1 αναδρομή

```
procedure Inorder (T);
begin
  while T<>empty do
  begin
    Inorder(leftchild(T)); Visit(T);
    T:=rightchild(T)
  end
end
```

Ενδοδιατεταγμένη διάσχιση χωρίς αναδρομή

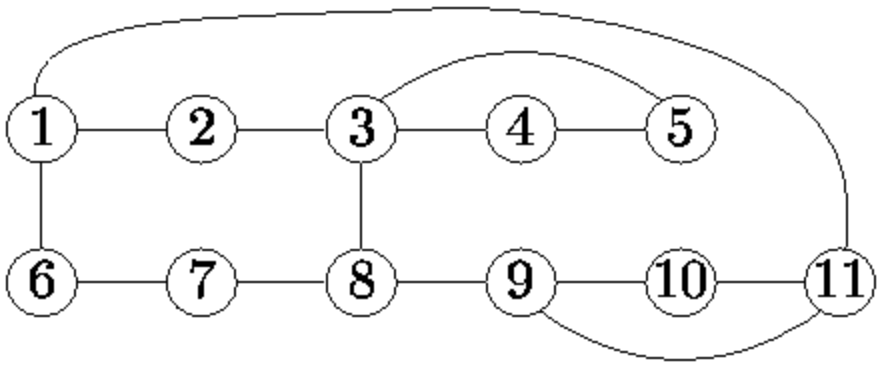
```
procedure Inorder (T);
(* Υποθέτουμε ότι έχουμε μια εικονική μεταβλητή T για την
   οποία ισχύει rightchild(T)=tree *)
begin
  stack:=empty; push(T);
  repeat
    while rightchild(T)<>empty do
    begin
      T:=rightchild(T);
      while leftchild(T)<>empty do
        begin push(T); T:=leftchild(T) end;
      Visit(T)
    end;
    pop(T); Visit(T)
  until stack=empty
end
```

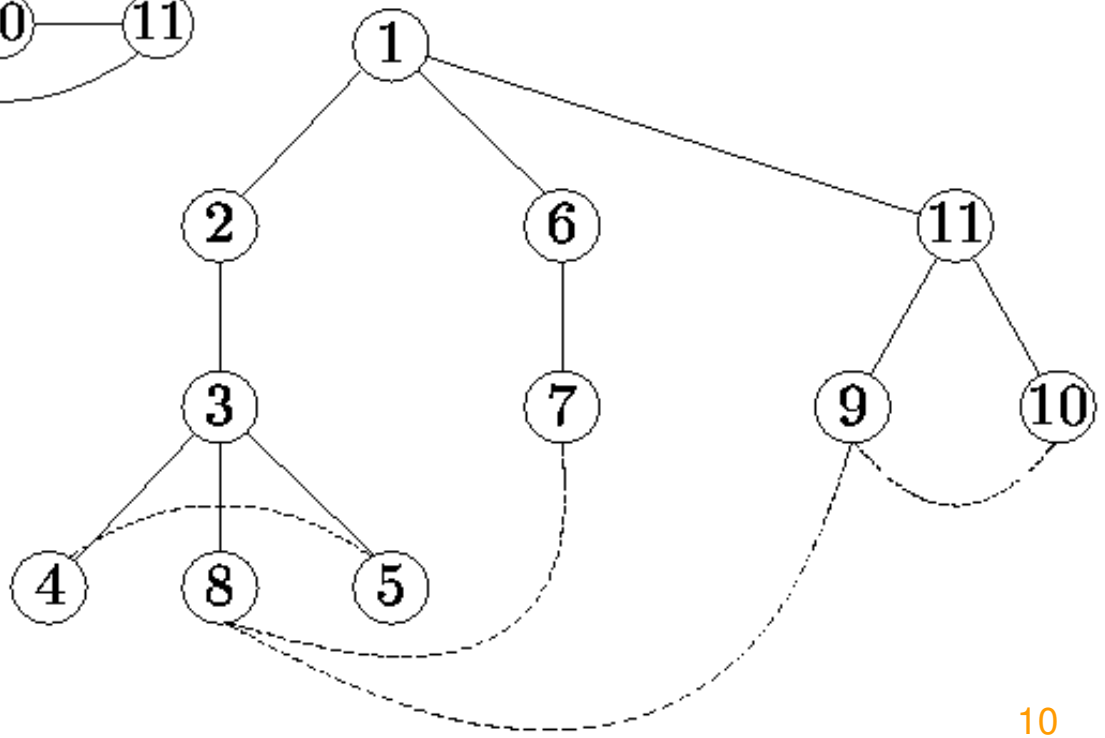
Διάσχιση Γράφων

Αποσκοπεί στην συστηματική επίσκεψη των κόμβων ενός γράφου ώστε να απαντάμε σε *Graph Accessibility Problems (GAP)* :

- Υπάρχει μονοπάτι από τον v στον u;
- Είναι ο γράφος ακυκλικός;
- Εύρεση συνεκτικών συνιστωσών (connected components).
- Εύρεση σημείων σύνδεσης (articulation points).

Αναζήτηση κατά Πλάτος (Breadth First Search - BFS)





Αλγόριθμος BFS

```
procedure bfs(v:vertex);
begin
  initialize queue with v; visited[v]:=true;
  repeat dequeue(u);
    for all vertices w adjacent to u do
      if not visited[w] then
        begin visited[w] := true; enqueue(w) end
  until queue is empty
end
```

Πίνακας Εκτέλεσης BFS

visited nodes	Queue
1	1
2	2
6	2-6
11	2-6-11
3	6-11-3
7	11-3-7
9	3-7-9
10	3-7-9-10
4	7-9-10-4
8	7-9-10-4-8
5	7-9-10-4-8-5
-	9-10-4-8-5
-	10-4-8-5
-	4-8-5
-	8-5
-	5
-	Ø

Πολυπλοκότητα BFS

- $O(n^2)$ με πίνακα γειτονίας.
- $O(n + e)$ με λίστες γειτονικών κορυφών.

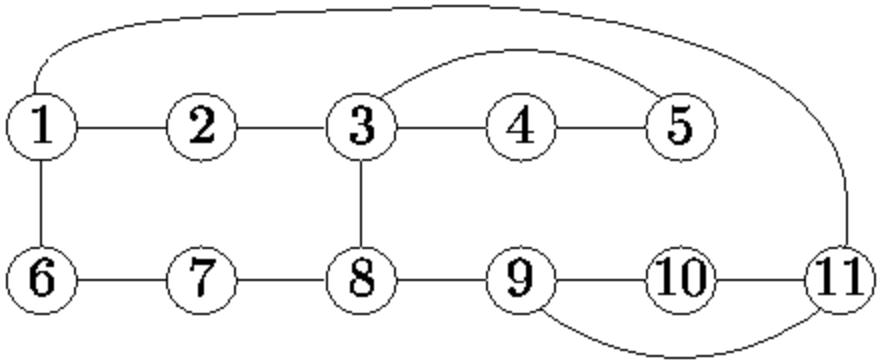
Εύρεση Συνεκτικών Συνιστωσών

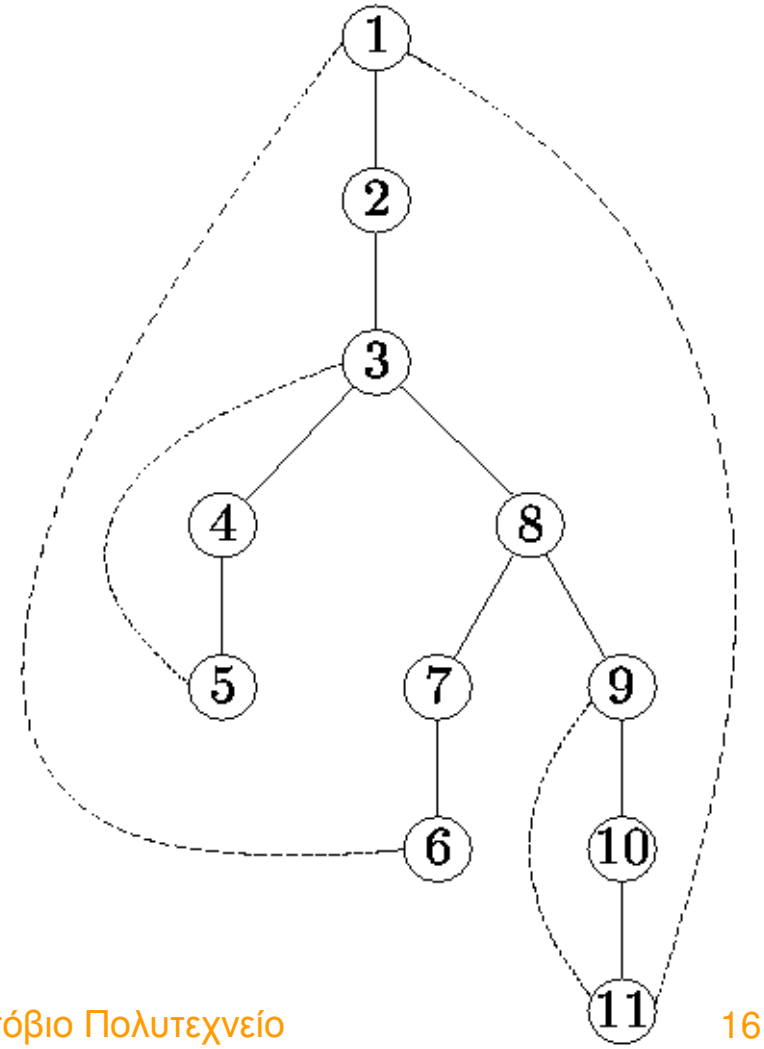
```
procedure bft(G);
begin
   initialize visited with false;
   for i:=1 to n do
      if not visited[i] then bfs(i)
end
```

Αναζήτηση κατά Βάθος (Depth First Search - DFS)

```
procedure dfs(v:vertex);
begin
   visited[v]:=true;
   for all vertices u adjacent to v do
      if not visited[u] then dfs(u)
end
```

Εκτέλεση DFS





D-Search

- Όμοιο με BFS
- Χρήση στοίβας αντί ουράς