

# *Μη Κατευθυνόμενα Γραφήματα II*

## **Αλγόριθμοι και Πολυπλοκότητα**

**Δημήτρης Λεονταρίδης**

# *Μη Κατευθυνόμενα Γραφήματα II*

- ✓ Διέλευση Γράφου
- ✓ Διάσχιση κατά Βάθος
- ✓ Διάσχιση κατά Πλάτος

# Διέλευση Γράφου

## ✓ Preorder (Προ-διατεταγμένη)

- Διαπέραση της ρίζας
- Αναδρομική διέλευση αριστερού υποδένδρου.
- Αναδρομική διέλευση δεξιού υποδένδρου.

## ✓ Postorder (Μετα-διατεταγμένη)

- Αναδρομική διέλευση αριστερού υποδένδρου.
- Αναδρομική διέλευση δεξιού υποδένδρου.
- Διαπέραση της ρίζας

## ✓ Inorder (Ενδο-διατεταγμένη)

- Αναδρομική διέλευση αριστερού υποδένδρου.
- Διαπέραση της ρίζας
- Αναδρομική διέλευση δεξιού υποδένδρου.

**Ερώτηση:** Ποια είναι η διαφορά μεταξύ της **Διέλευσης** και της **Αναζήτησης** (BFS/DFS)

- **Διέλευση** -> Ταξινόμηση, Αντιγραφή, Αναδιάταξη εκφράσεων (μαθηματικών πράξεων)
- **Αναζήτηση** -> Εύρεση μονοπατιού προς μία κορυφή, η οποία είναι ο τελικός προορισμός. Η Αναζήτηση σταματάει όταν φτάσουμε στον προορισμό.

# Αναζήτηση - Διάσχιση

## ✓ Διάσχιση κατά Βάθος (DFS)

- Επίσκεψη κορυφής
- Επίσκεψη **ενός** ανεξερεύνητου γείτονα της κορυφής

## ✓ Διάσχιση κατά Πλάτος (BFS)

- Επίσκεψη κορυφής
- Επίσκεψη **όλων** των ανεξερεύνητων γειτόνων της κορυφής

**Αποτέλεσμα:** Παραγωγή Γεννητορικού δένδρου, δίχως κύκλους

# Αναζήτηση - Διάσχιση

## ✓ DFS Αλγόριθμος (με Αναδρομή)

```
main () {  
    for v ∈ V  
        visited[v] ← false;  
    for v ∈ V  
        if not visited[v] then  
            explore(v, visited);  
}
```

```
explore (v, visited) {  
    visited[v] ← true;  
    for u ∈ N(v)  
        if not visited[u] then  
            explore(u, visited);  
}
```

# Αναζήτηση - Διάσχιση

## ✓ DFS Αλγόριθμος με Στοίβα (*χωρίς Αναδρομή*)

```
main () {  
    create Q;  
    clock ← 0;  
    for v ∈ V {  
        preorder[v] ← 0;  
        visited[v] ← false;  
    }  
    for v ∈ V  
        if not visited[v] then  
            push (v, Q);  
            explore(Q, visited, preorder, clock);  
}
```

```
explore (Q, visited preorder, clock) {  
    while |Q| ≠ 0 {  
        v ← pop(Q);  
        if not visited[v] then  
            visited[v] ← true;  
            preorder[v] ← ++ clock  
            for u ∈ N(v)  
                if not visited[u] then  
Εισαγωγή των στοιχείων από μπροστά  
μέρος της στοίβας  
                append(u, Q);  
    }  
}
```

# Αναζήτηση - Διάσχιση

## ✓ BFS Αλγόριθμος με Στοιίβα (*χωρίς Αναδρομή*)

```
main () {  
    create Q;  
    clock ← 0;  
    for v ∈ V {  
        preorder[v] ← 0;  
        visited[v] ← false;  
    }  
    for v ∈ V  
        if not visited[v] then  
            push (v, Q);  
            explore(Q, visited, preorder, clock);  
}
```

```
explore (Q, visited preorder, clock) {  
    while |Q| ≠ 0 {  
        v ← pop(Q);  
        if not visited[v] then  
            visited[v] ← true;  
            preorder[v] ← + + clock  
            for u ∈ N(v)  
                if not visited[u] then  
Εισαγωγή των στοιχείων από πίσω μέρος  
της στοίβας  
                    push(u, Q);  
    }  
}
```