

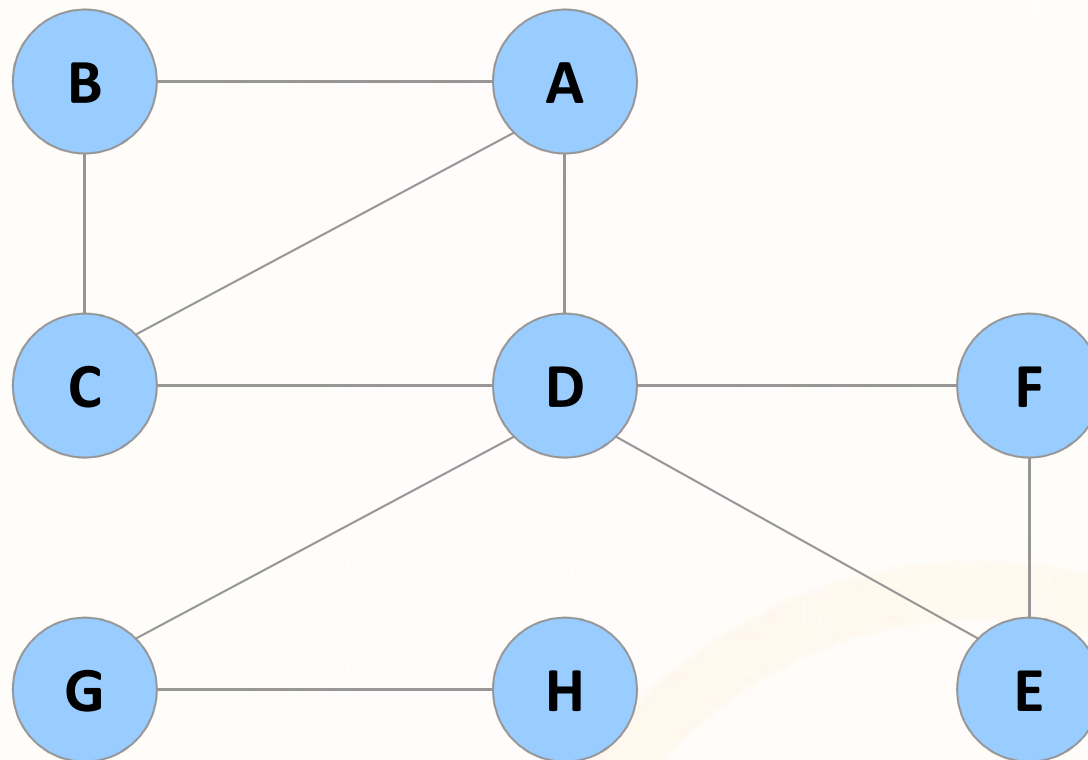


Εργαστήριο 11: Γράφοι

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

- Διερεύνηση σε πλάτος και σε βάθος (Depth First Search & Breadth First Search)
- Απορίες για άσκηση 5

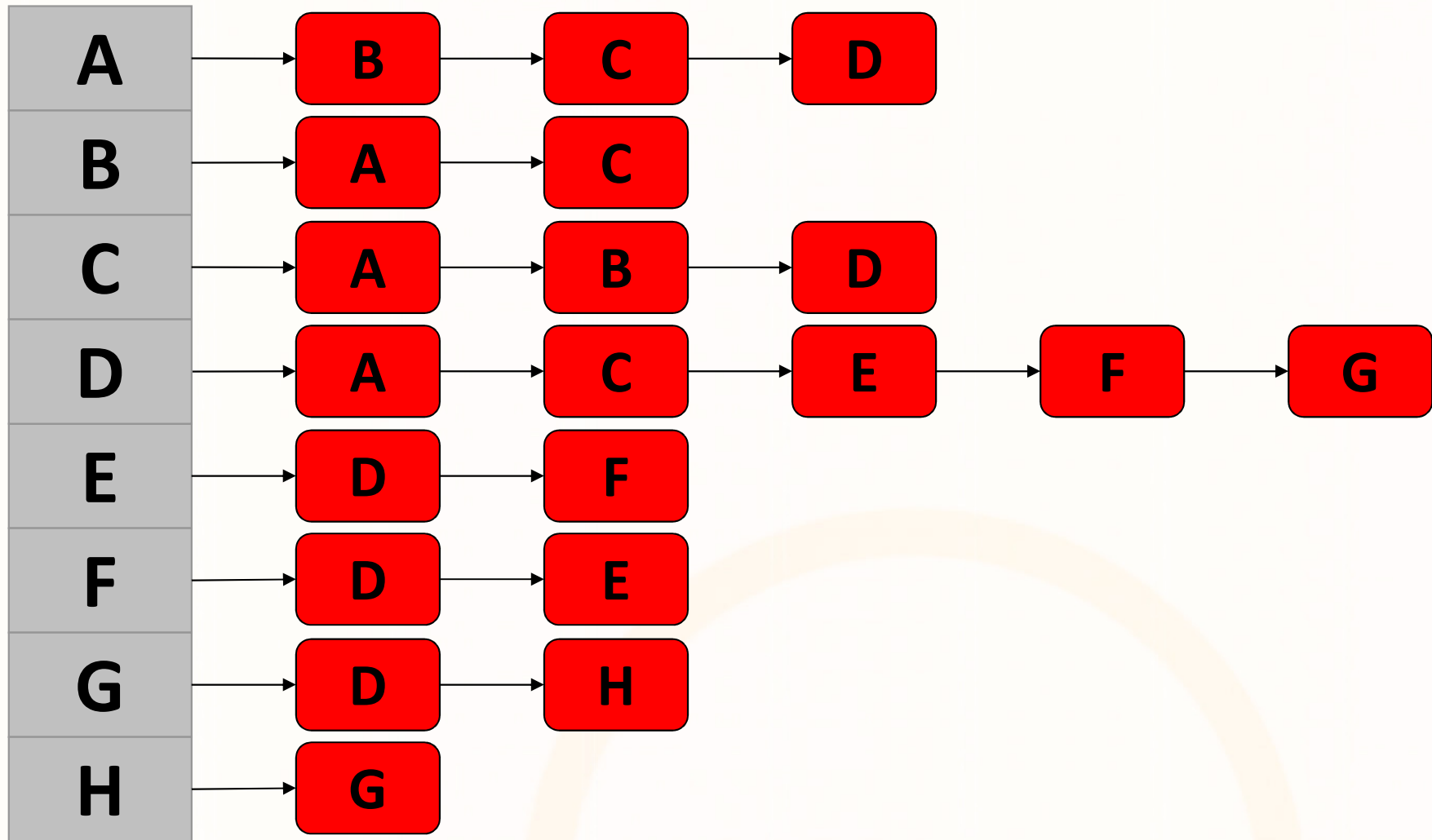
Παράδειγμα Γράφου



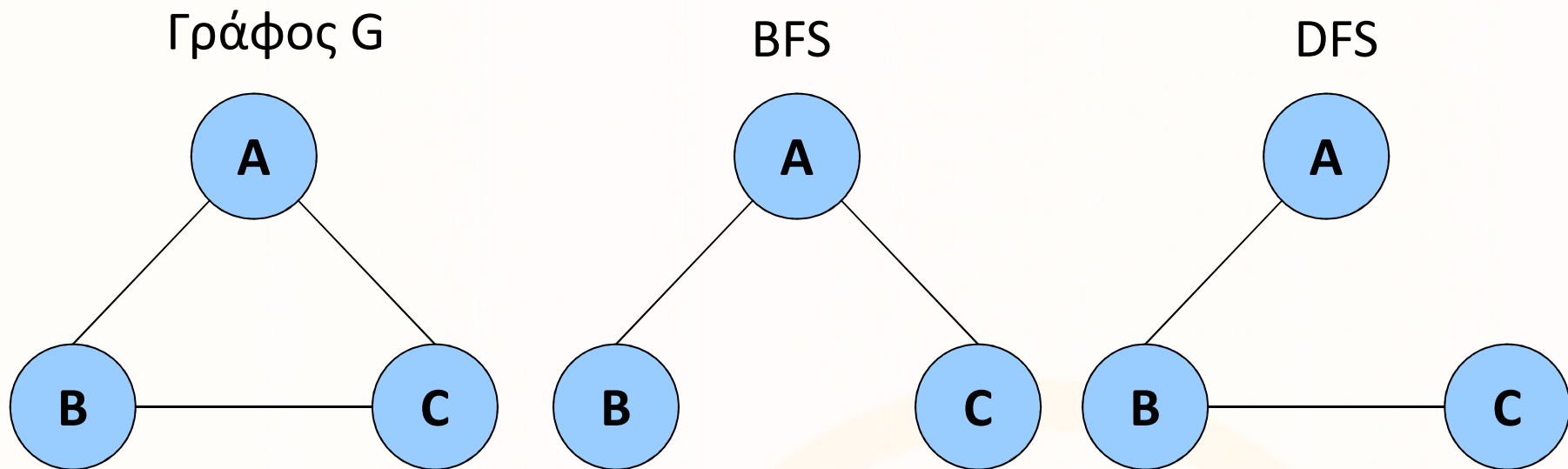
Πίνακας Γειτνίασης

| | A=0 | B=1 | C=2 | D=3 | E=4 | F=5 | G=6 | H=7 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| A=0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| B=1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| C=2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| D=3 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| E=4 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| F=5 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| G=6 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| H=7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

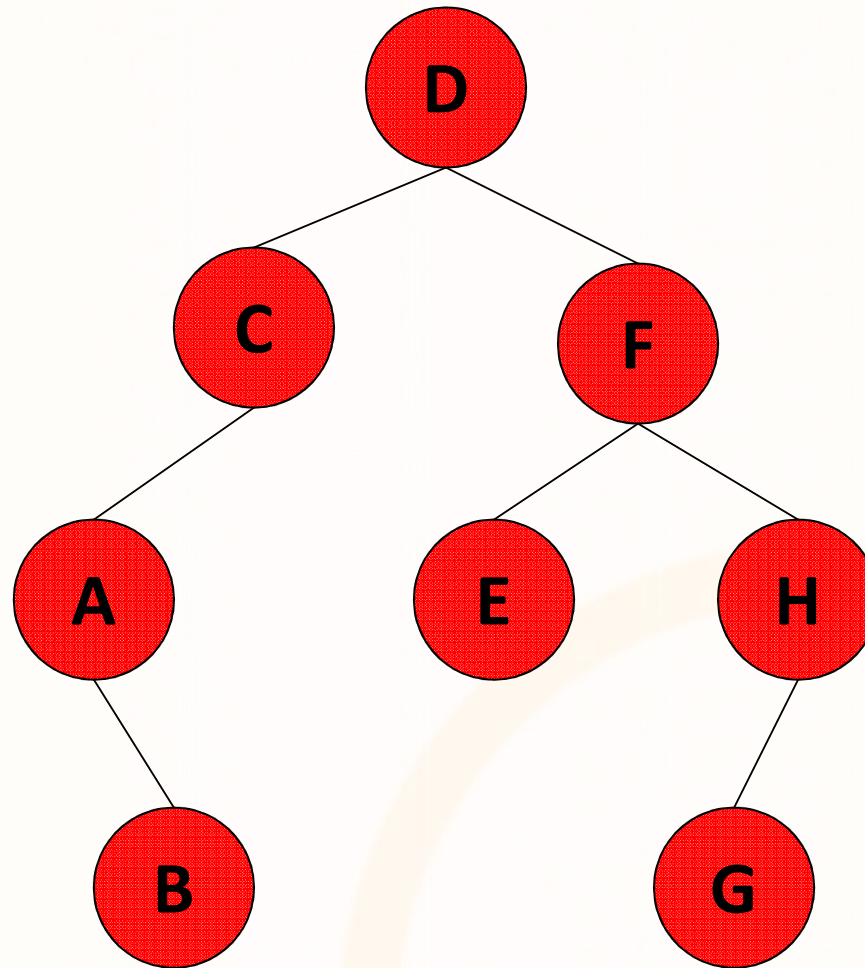
Λίστα Γειτνίασης



Συζήτηση



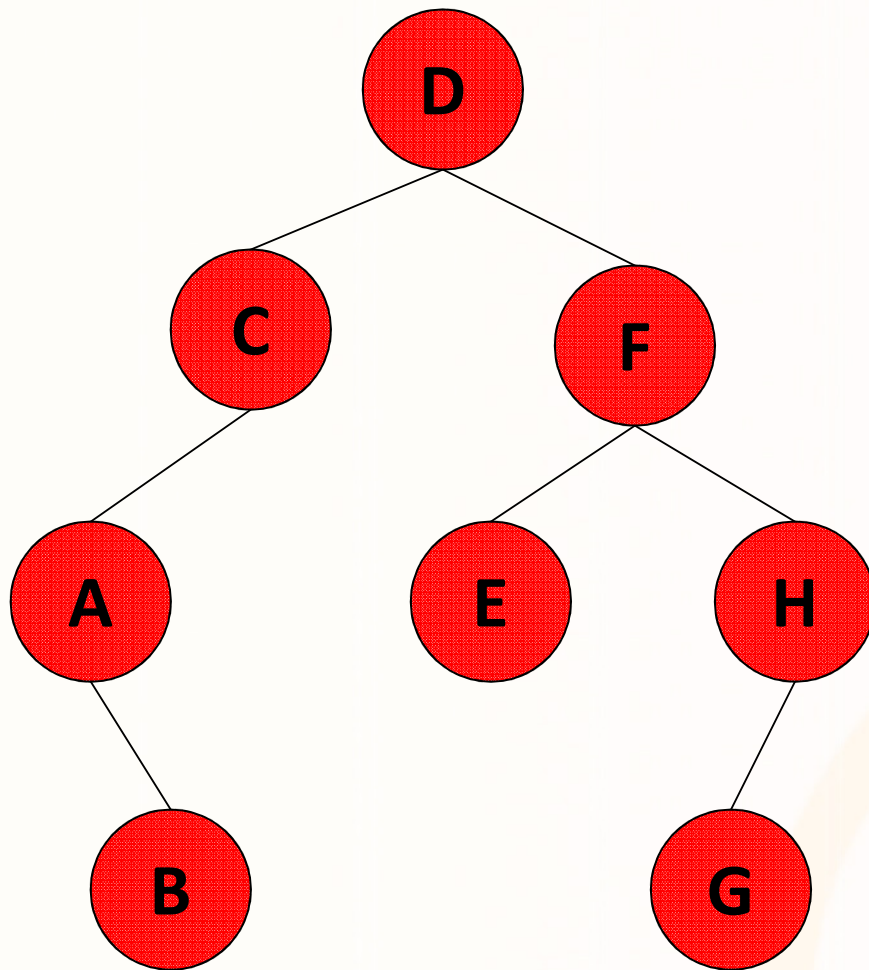
Παράδειγμα Εκτέλεσης DFS



Έξοδος Διαδικασίας

D
C
A
B
F
E
H
G

Παράδειγμα Εκτέλεσης BFS



Ουρά Q

D

C,F

F, A

A, E, H

E, H, B

H, B

B, G

G

{}

Έξοδος Διαδικασίας

D

C

F

A

E

H

B

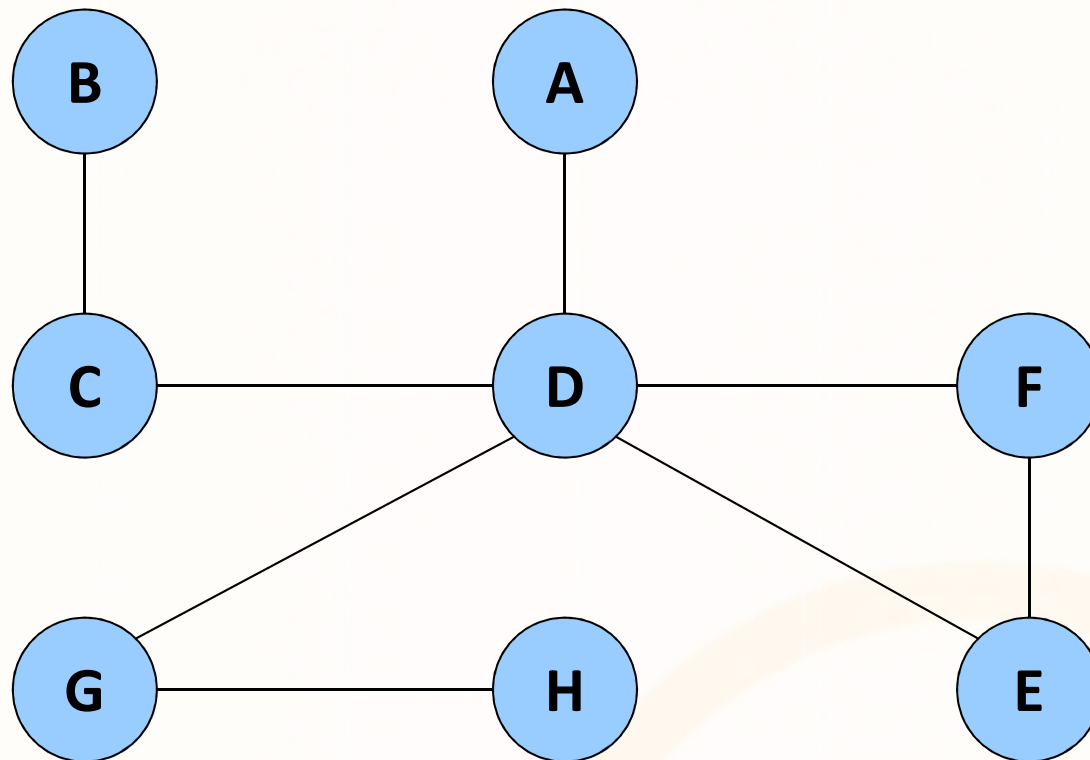
G

Δομές για υλοποίηση

- Πίνακας Γειτνίασης
 - Πίνακας [8]x[8]
 - Αρχικοποίηση βάση γράφου επόμενης διαφάνειας
- Ουρά για BFS
- Κατεβάστε το αρχείο με τον σκελετό της άσκησης

<http://www.cs.ucy.ac.cy/courses/EPL035/labs/2012F.EPL035.labs.11.dfs.bfs.txt>

Γράφος για έλεγχο



Συναρτήσεις για υλοποίηση

- *void DFS(Graph G, Vertex v):* ξεκινάει την *διαδικασία διερεύνησης σε βάθος* από τον κόμβο v του γράφου G
- *void BFS(Graph G, Vertex v):* ξεκινάει την *διαδικασία διερεύνησης σε πλάτος* από τον κόμβο v του γράφου G

Λύσεις (DFS)

```
void DFS(GRAPH* g, int vertex){
    char c = 65+vertex;
    printf("%c\n", c);
    g->visited[vertex]=1;
    for(int i=0; i<g->vertices; i++){
        if(g->visited[i]==0
            && g->matrix[vertex][i]==1)
            DFS(g, i);
    }
}
```

Λύσεις (BFS)

```
void BFS(GRAPH* g, int vertex){
    char c;
    QUEUE Q;
    MakeEmptyQueue(&Q);
    int v;
    Enqueue(&Q, vertex);
    while(!IsEmptyQueue(&Q)){
        v = Dequeue(&Q);
        c = 65+v;
        printf("%c\n", c);
        g->visited[v]=1;
        for(int i=0; i<g->vertices; i++){
            if(g->visited[i]==0
                && g->matrix[v][i]==1) {
                g->visited[i] = 1;
                Enqueue(&Q, i);
            }
        }
    }
}
```