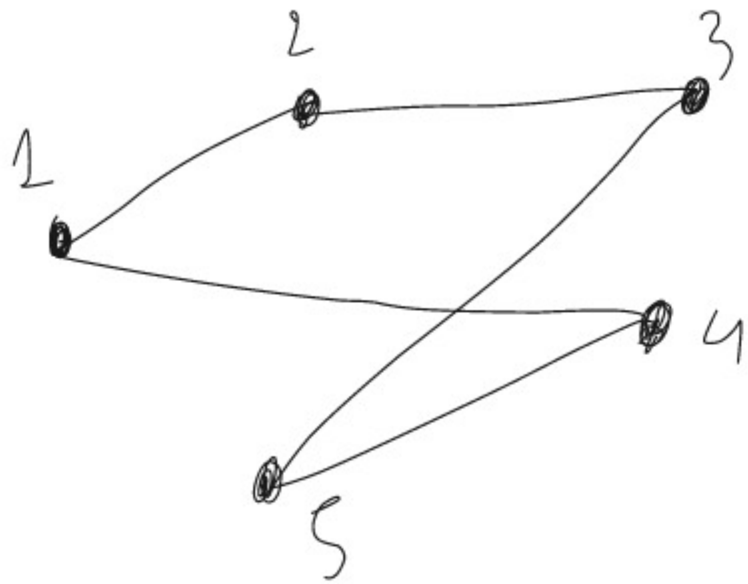
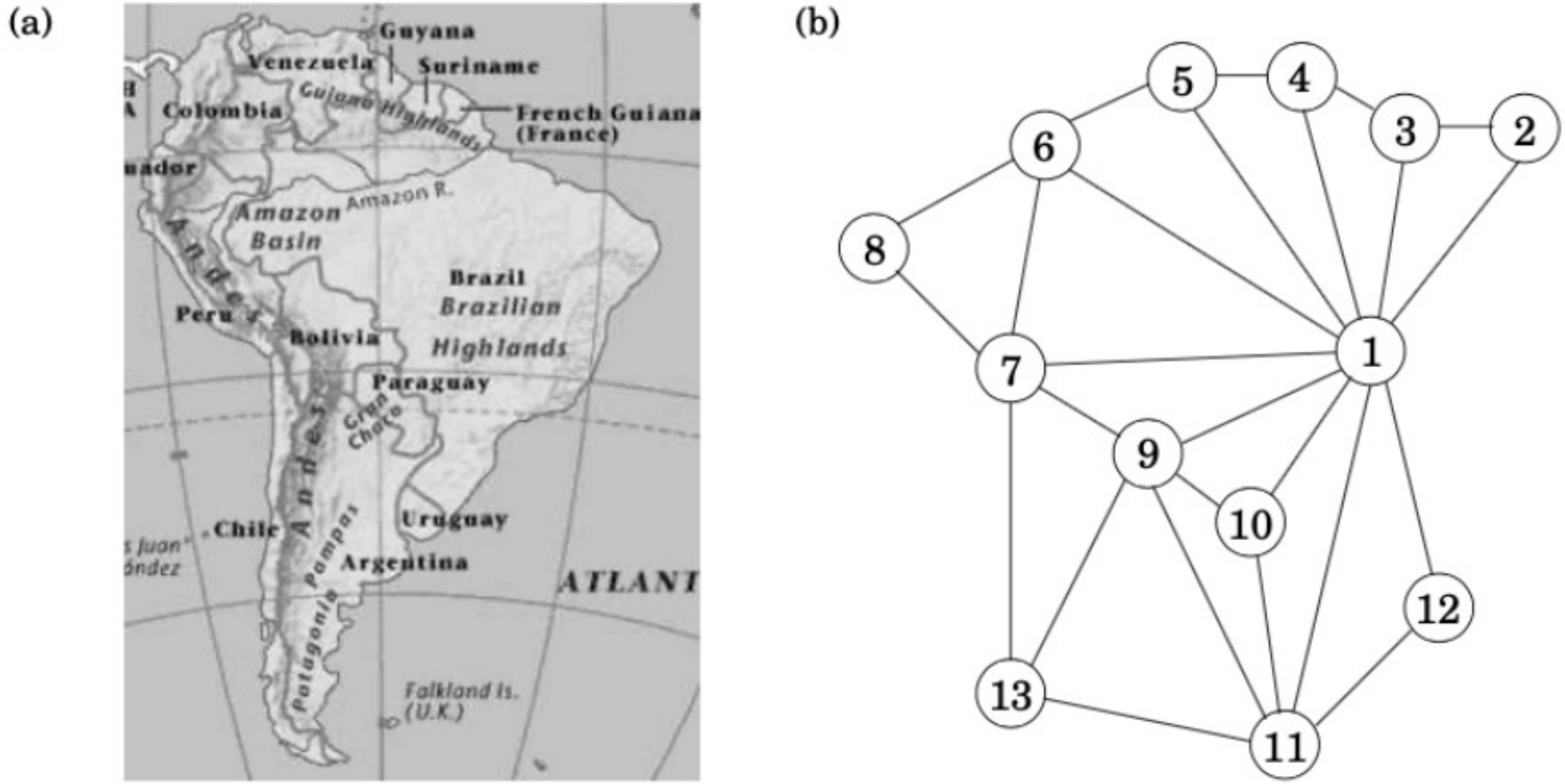
$G = (V, E)$

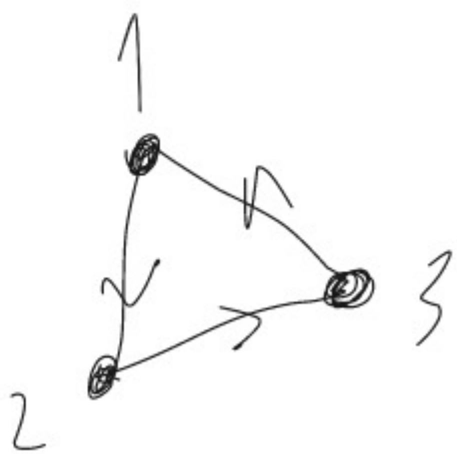
κορυφών ακμών



$V = \{1, 2, 3, 4, 5\}$

$E = \{\{1,2\}, \{2,3\}, \{5,3\}, \{4,1\}, \{5,4\}\}$



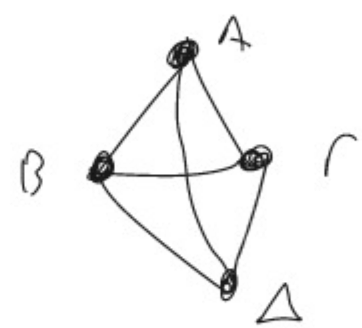


$V = \{1, 2, 3\}$

$E = \{(1,2), (2,3), (3,1)\}$

Αναπαράσταση γραφημάτων:

• Πίνακας γειτνίασης: $(n = |V|,\ m = |E|)$



$$G = \begin{array}{c|cccc} & A & B & \Gamma & \Delta \\ \hline A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ \Gamma & 1 & 1 & 0 & 1 \\ \Delta & 1 & 1 & 1 & 0 \end{array}$$

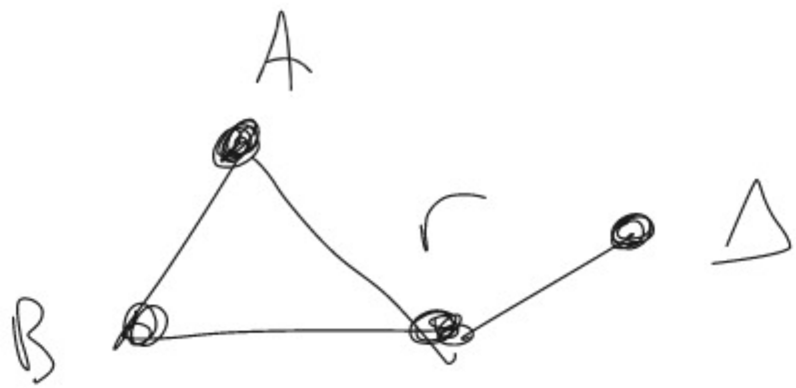
Έστω $G=(V,E)$ με n κορυφές ο πίνακας γειτνίασης είναι ένας $n \times n$ πίνακας A με:

$$A[i,j] = \begin{cases} 1, & \text{αν υπάρχει ακμή } u_i, u_j \\ 0, & \text{αλλιώς.} \end{cases}$$

όπου $V = \{u_1, \dots, u_n\}$

χώρος $O(n^2)$

• Λίστα γειτνίασης;



A □ — [B, Γ]

B □ — [A, Γ]

Γ □ — [A, B, Δ]

Δ □ — [Γ]

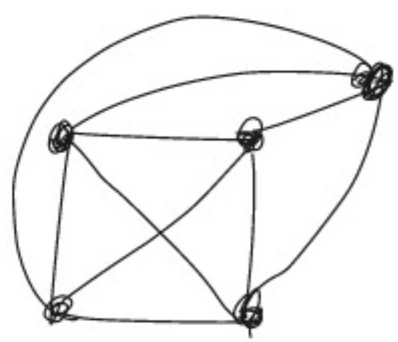
Χώρος $O(m)$

$n = 3$

$m = 0$

$n = 3$

$m = 2 \quad (n-1)$



n

$$m = \binom{n}{2} = \frac{n^2 - n}{2} = O(n^2)$$

Αναζήτηση κατά Βάθος (DFS):

explore(G,v): (G,v,cc)

Είσοδος: Γράφημα $G=(V,E)$, $v \in V$

Έξοδος: Τίποτα (το visited(u) παίρνει τιμή true για κάθε u προσπελάσιμο από την v)

```
S = [v]        (στοίβα που περιέχει μόνο την v)
while S ≠ ∅
    v = pop(S)
    if not visited(v)
        visited(v) = true | ccnum(v) = cc
        for every (v,u) ∈ E  (ή {v,u})
            push(S,u)
```

explore(G,v):

Είσοδος: —//—

Έξοδος: —//—

```
visited(v) = true
for every (v,u) ∈ E
    if not visited(u)
        explore(G,u)
```

Χρόνος $O(|E'|)$

E' το πλήθος ακμών της σ.σ. που ανήκει η v.

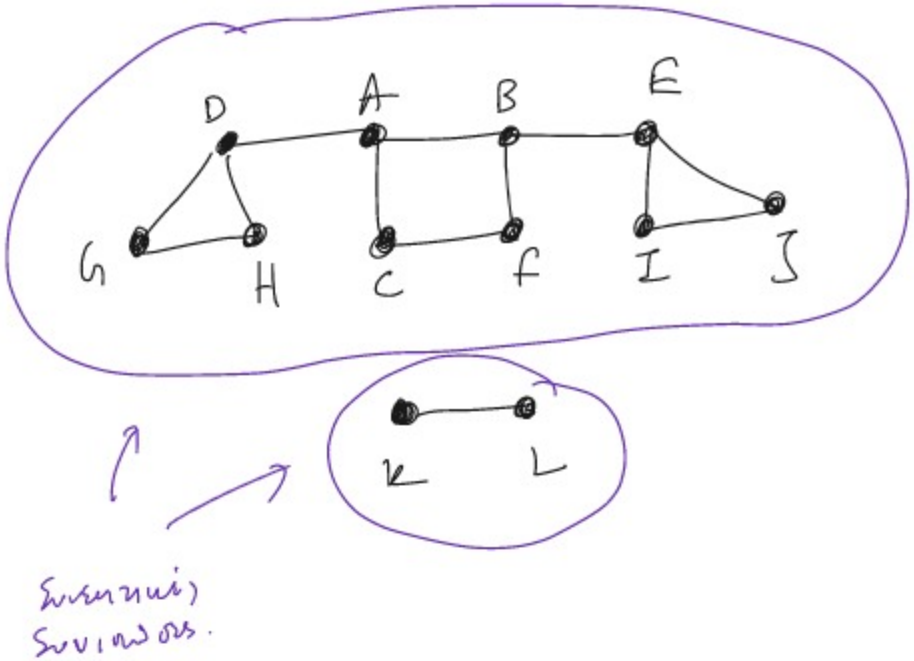
DFS(G):

Είσοδος: Γράφημα $G=(V,E)$

Έξοδος: Τίποτα (το visited(u) παίρνει τιμή true για κάθε $u \in V$)

```
for every v ∈ V
    visited(v) = false
for every v ∈ V
    if not visited(v)
        explore(G,v)
```

$O(|V|+|E|)$

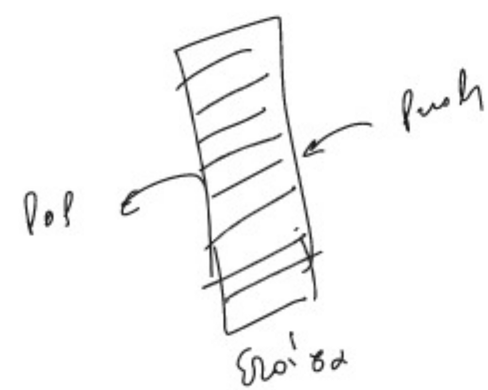


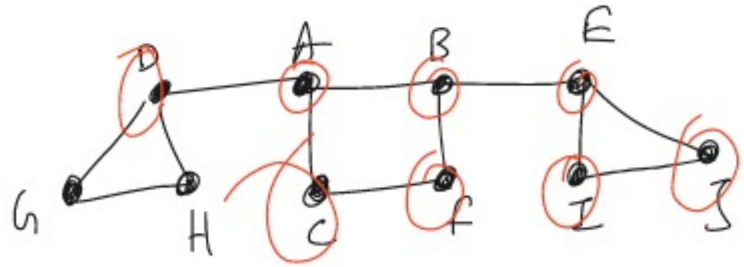
A — B, C, D
B — A, E, F
C — A, F
D — A, G, H
E — B, I, J
F — B, C
G — D, H
H — D, G
I — E, J
J — E, I
K — L
L — K

- Κι μνήμη:

$$\text{visited}(u) = \begin{cases} \text{true}, & \text{έχουμε επισκεφτεί τον } u \\ \text{false}, & \text{αλλιώς.} \end{cases}$$

- Υλικά: Στοίβα





<u>explore (G,v):</u>

Είσοδος: Γράφημα $G=(V,E)$, $v \in V$

Έξοδος: Τίποτα (το visited(u) γίνεται true αν το u προσπελάσιμο από τον v)

for every $v \in V$ visited(v) = false

$S=[v]$ (στοίβα που περιέχει μόνο τον v)

while $S \neq \emptyset$

$v = pop(S)$

if not visited(v)

visited(v) = true

for every $(v,u) \in E$ (ή $\{v,u\}$)

push (S,u)

*

~~E~~ visited (E) = true

~~F~~ visited (F) = true

~~B~~ visited (B) = true

~~C~~

D

~~A~~ visited (A) = true

~~A~~

~~A~~

~~B~~

~~C~~ visited (C) = true

**

**

~~F~~

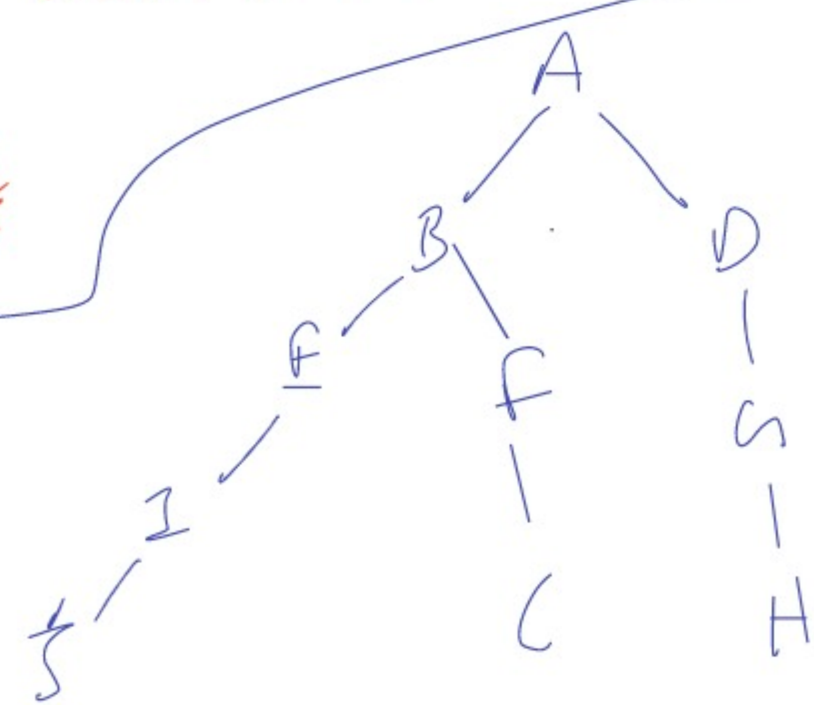
~~J~~

~~J~~ visited (J) = true

~~I~~ visited (I) = true

~~J~~

*



ΔΟ ο explore επισκέπτεται κάθε κορυφή που έχει απόσταση k από την v.

Επαγωγή:

Βάση: k=0 προφανώς ισχύει

Ε.Υ.: Ισχύει για k

Ε. Βήμα: Έστω ότι η κορυφή u που απέχει απόσταση k+1 από την v έχει visited(u) = false μετά το τέλος του explore(G, v).

Η απόσταση της w από την v είναι k άρα από Ε.Υ. άρα visited(w) = true στο τέλος της explore(G, v). Άρα η u θα μπει στην στοίβα, επειδή (και όσο κοιτάμε θα έχουμε pop(S) άρα) η τιμή του visited(u) θα γίνει true. Άτοπο.

DFS(G):

Είσοδος: γράφημα $G=(V,E)$

Έξοδος: Τίποτα (το visited(u) παίρνει τιμή true για κάθε $u \in V$)

cc=0

```
for every v ∈ V
    visited(v) = false
for every v ∈ V
    if not visited(v)
        cc = cc+1
        explore(G,v)
```

Αλγόριθμος κατά Πλάτος (BFS):

explore (G,v):

Είσοδος: —— // ——

Έξοδος: —— // ——

$Q = [v]$ (ουρά που περιέχει μόνο τον v)

```
while Q ≠ ∅
    v = eject(Q)
    visited(v) = true
    for every (v,u) ∈ E
        if not visited(u)
            inject(Q,u).
```

Χρόνος $O(|E|)$

$O(|V|+|E|)$

BFS (G):

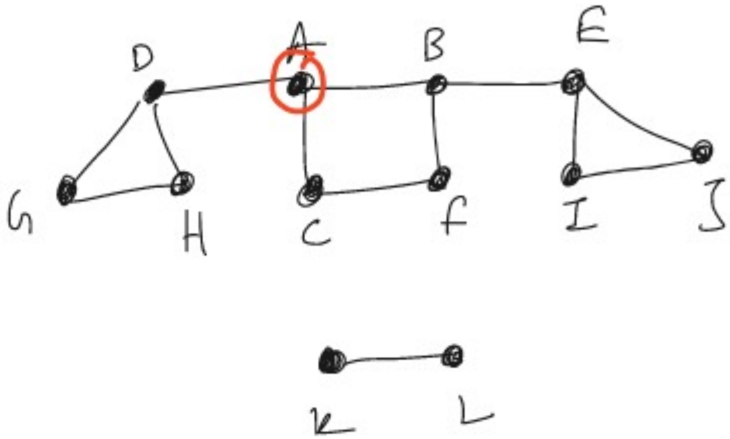
Είσοδος: —— // ——

Έξοδος: —— // ——

```
for every v ∈ V
    visited(v) = false

for every v ∈ V
    if not visited(v)
        explore(G,v)
```

miro

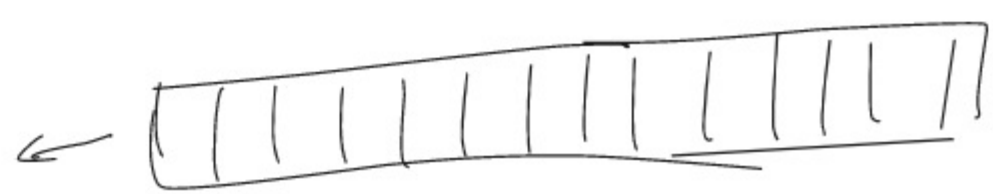


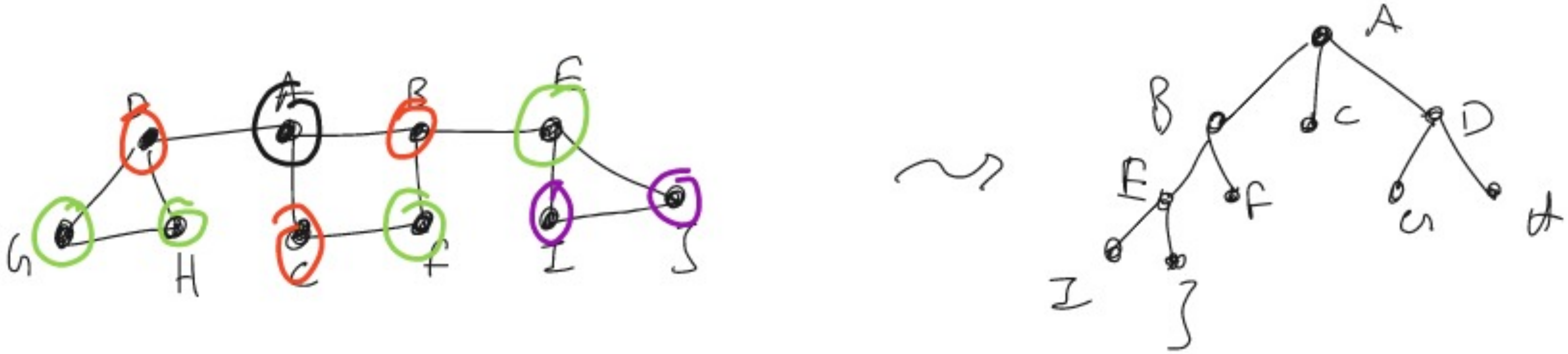
A – B, C, D
B – A, E, F
C – A, F
D – A, G, H
E – B, I, J
F – B, C
G – D, H
H – D, G
I – E, J
J – E, I
K – L
L – K

• Κόμβοι:

$$\text{visited}(u) = \begin{cases} \text{true}, & \text{έχουμε επισκεφθεί τον } u \\ \text{false}, & \text{αλλιώς} \end{cases}$$

• Ουρά:

eject  inject



D
A
B
E
G
H
C
F
I
J
A
B
C
D
E
F
G
H
I
J