

Camp Προετοιμασίας
Πανελλήνιος Διαγωνισμός Πληροφορικής
Αθήνα, 11 - 15 Μαρτίου 2013

ΓΡΑΦΟΙ I - BFS - DFS

Μάκης Αρσένης
Δημήτρης Τερζόπουλος

Επιμέλεια Διαφανειών: Δημήτρης Τερζόπουλος

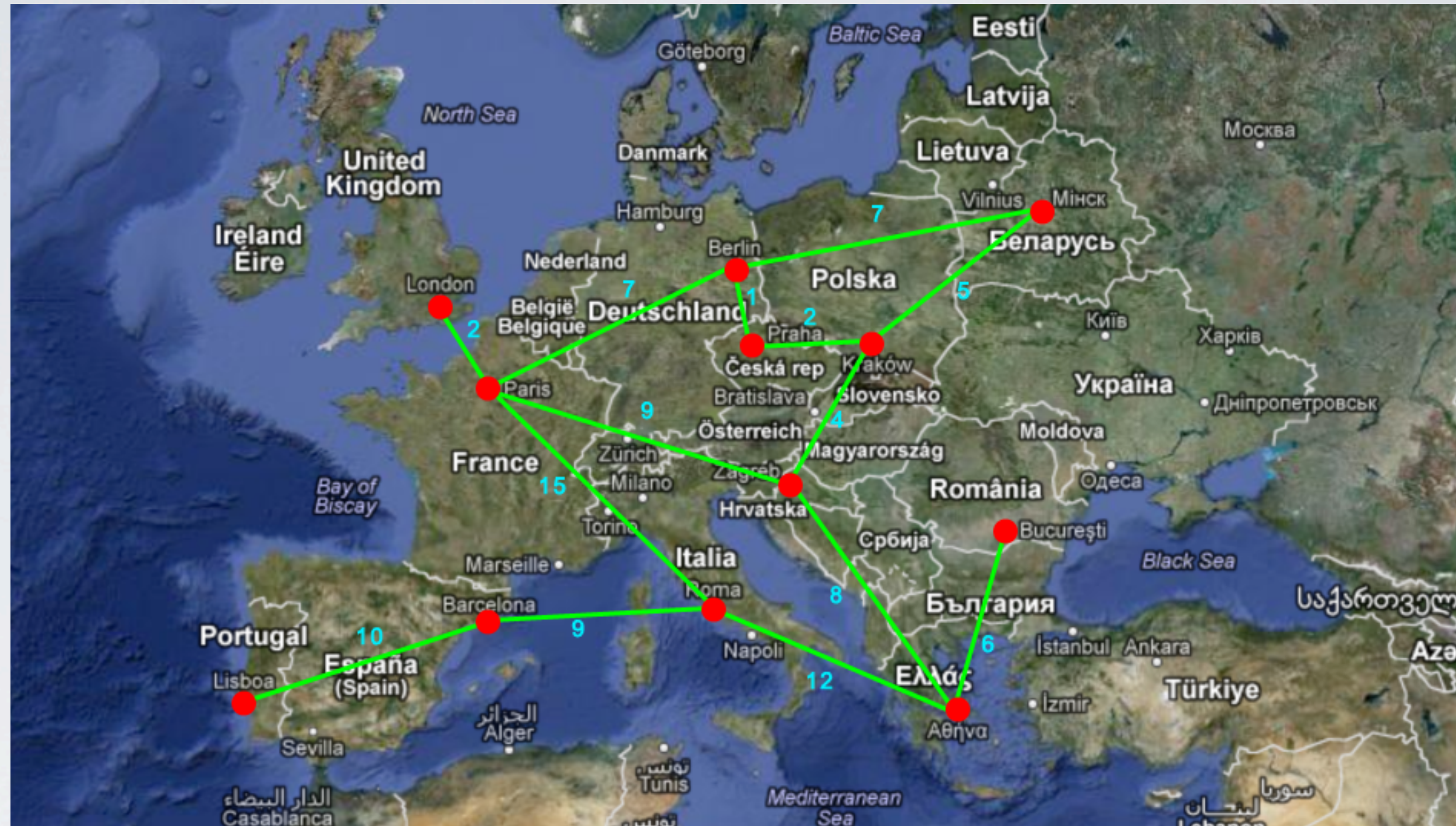




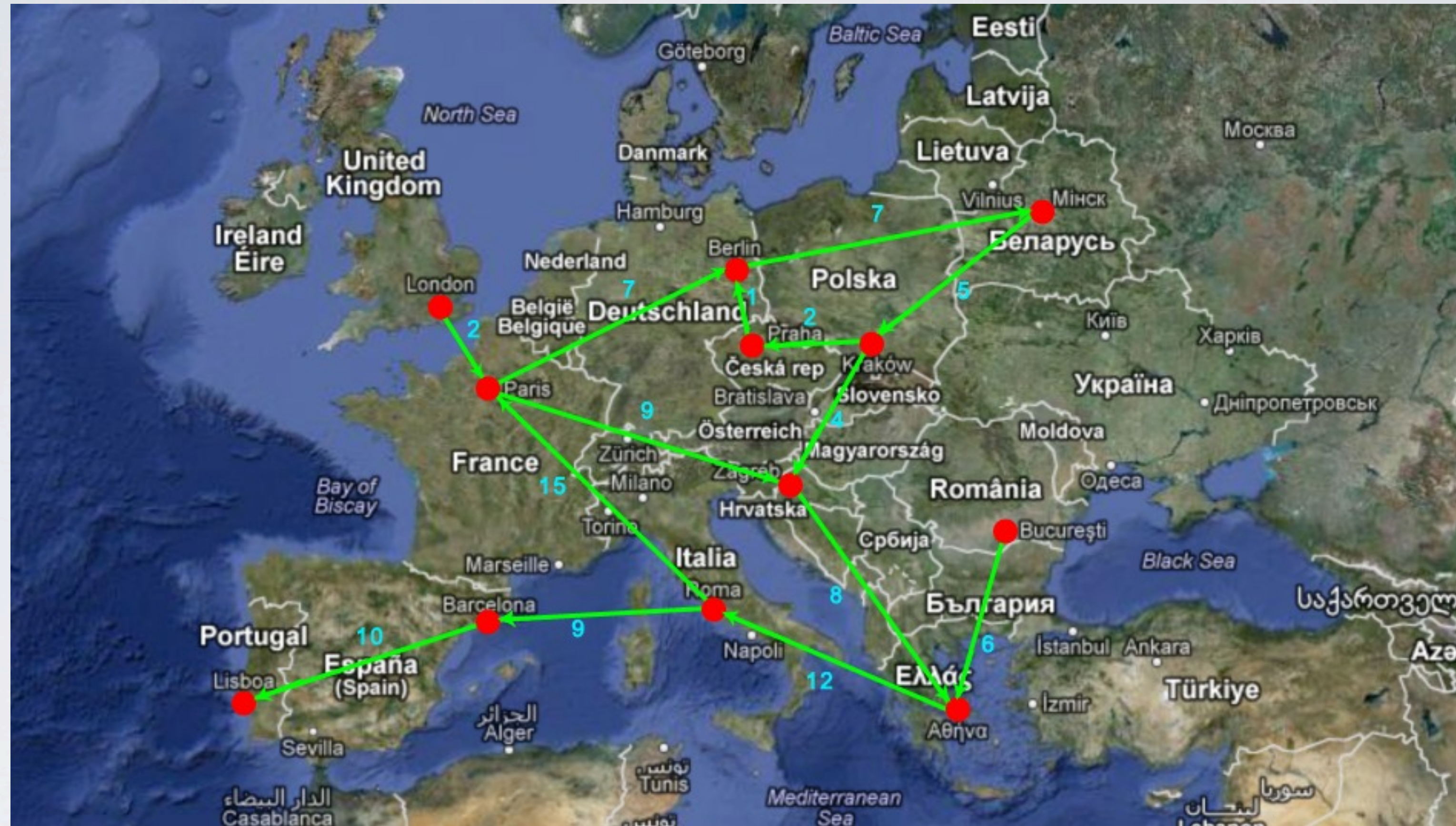
KOMBOI - VERTICES - NODES (V)



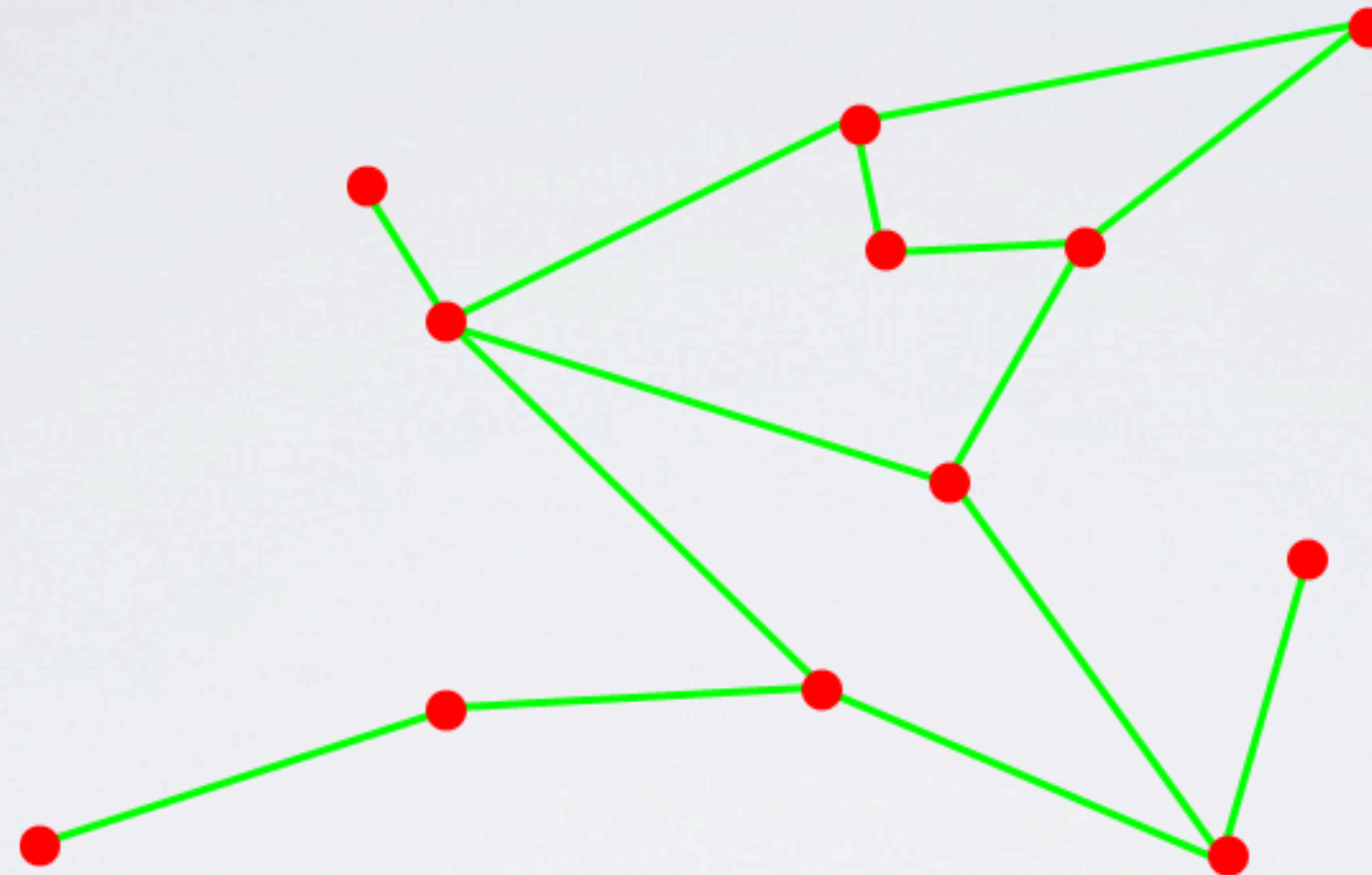
AKMEΣ - EDGES (E)



ΒΑΡΗ - ΑΠΟΣΤΑΣΕΙΣ (WEIGHTS)

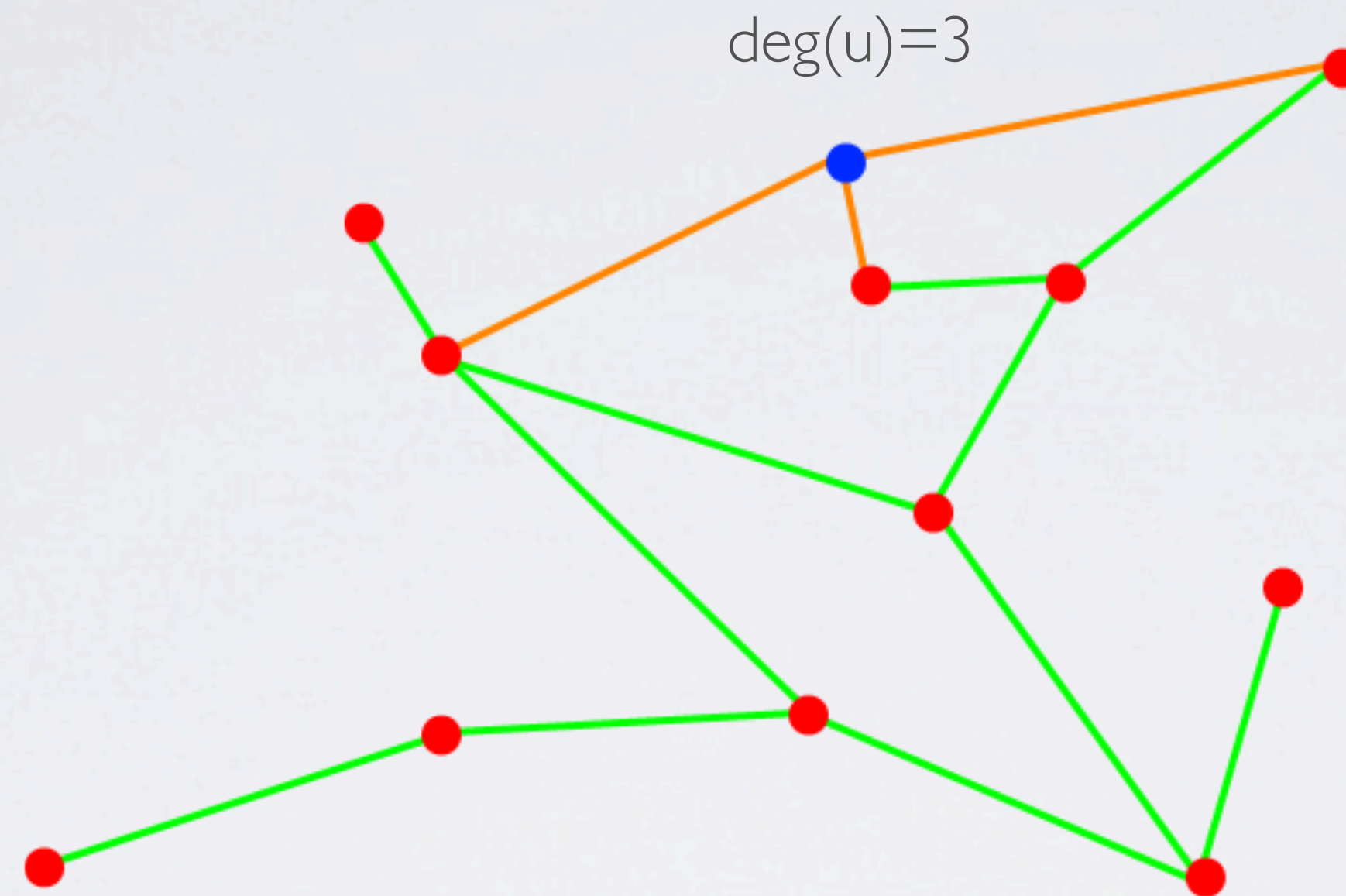


ΚΑΤΕΥΘΥΝΟΜΕΝΟΣ ΓΡΑΦΟΣ



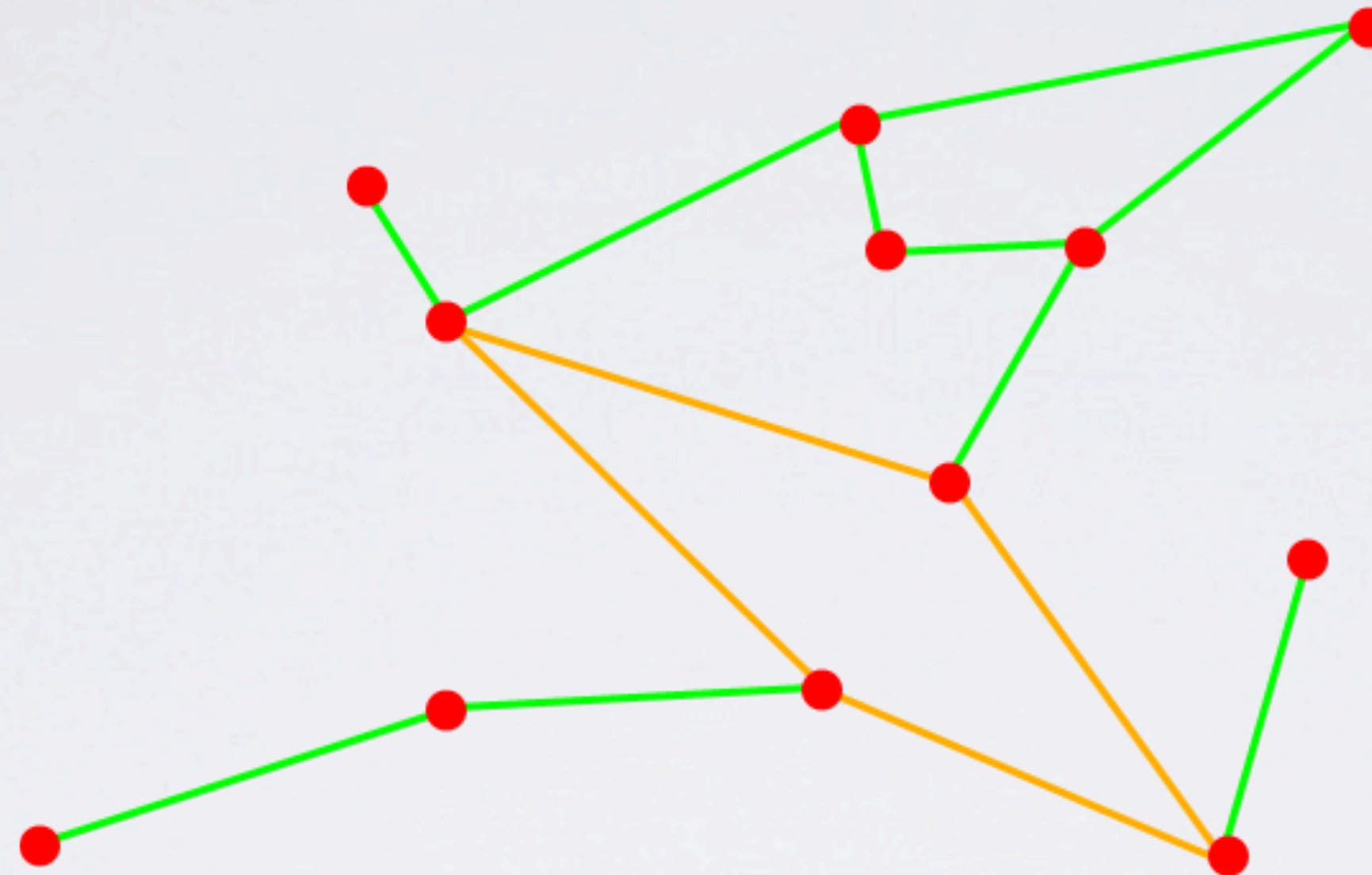
ΓΡΑΦΟΣ

$$G(V, E)$$



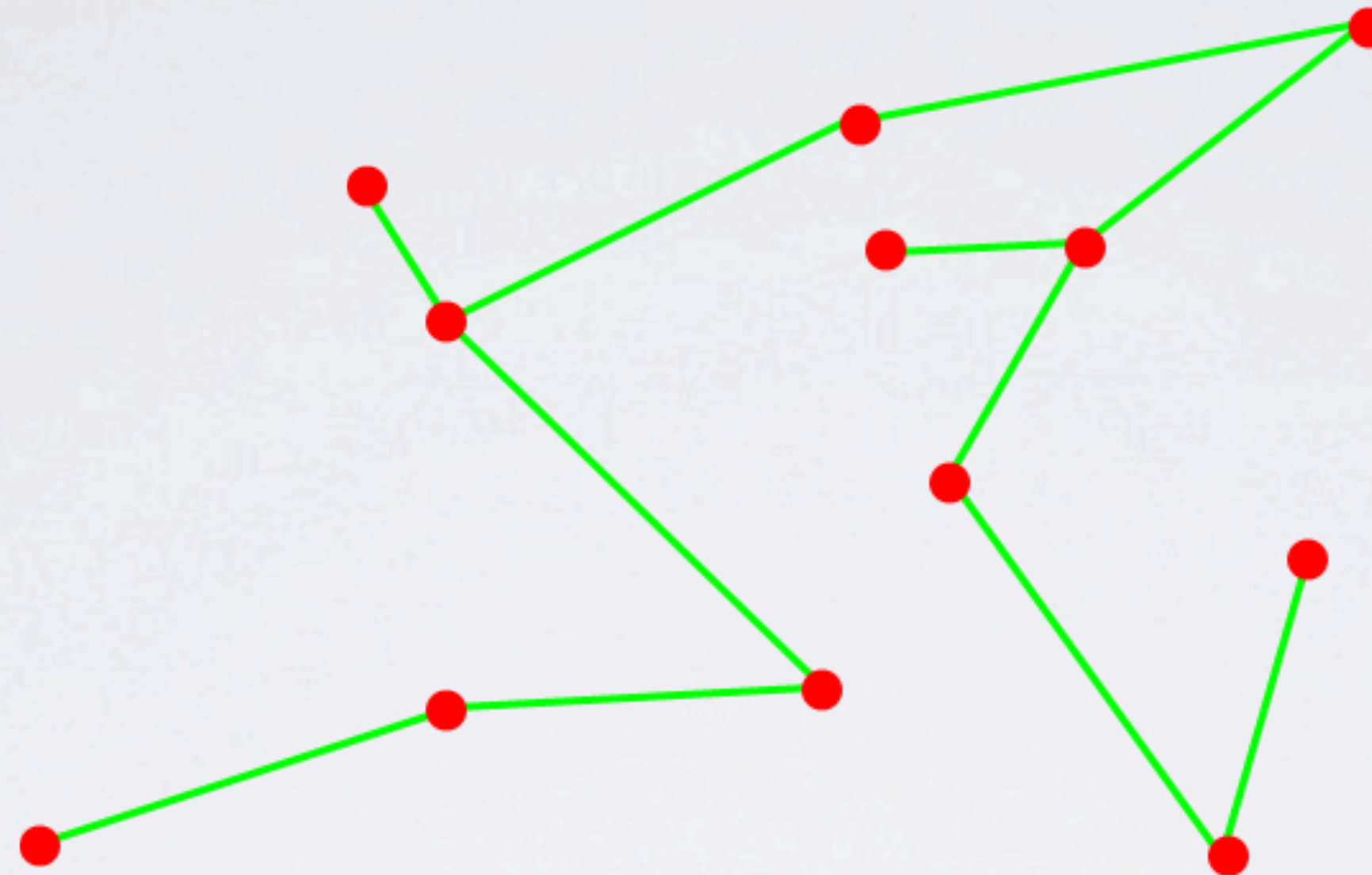
ΒΑΘΜΟΣ ΚΟΡΥΦΗΣ

- Undirected Graph: $\text{deg}(u)=2|E|$
- Directed Graph: $\text{deg}_{\text{incoming}}(u)$, $\text{deg}_{\text{outgoing}}(u)$



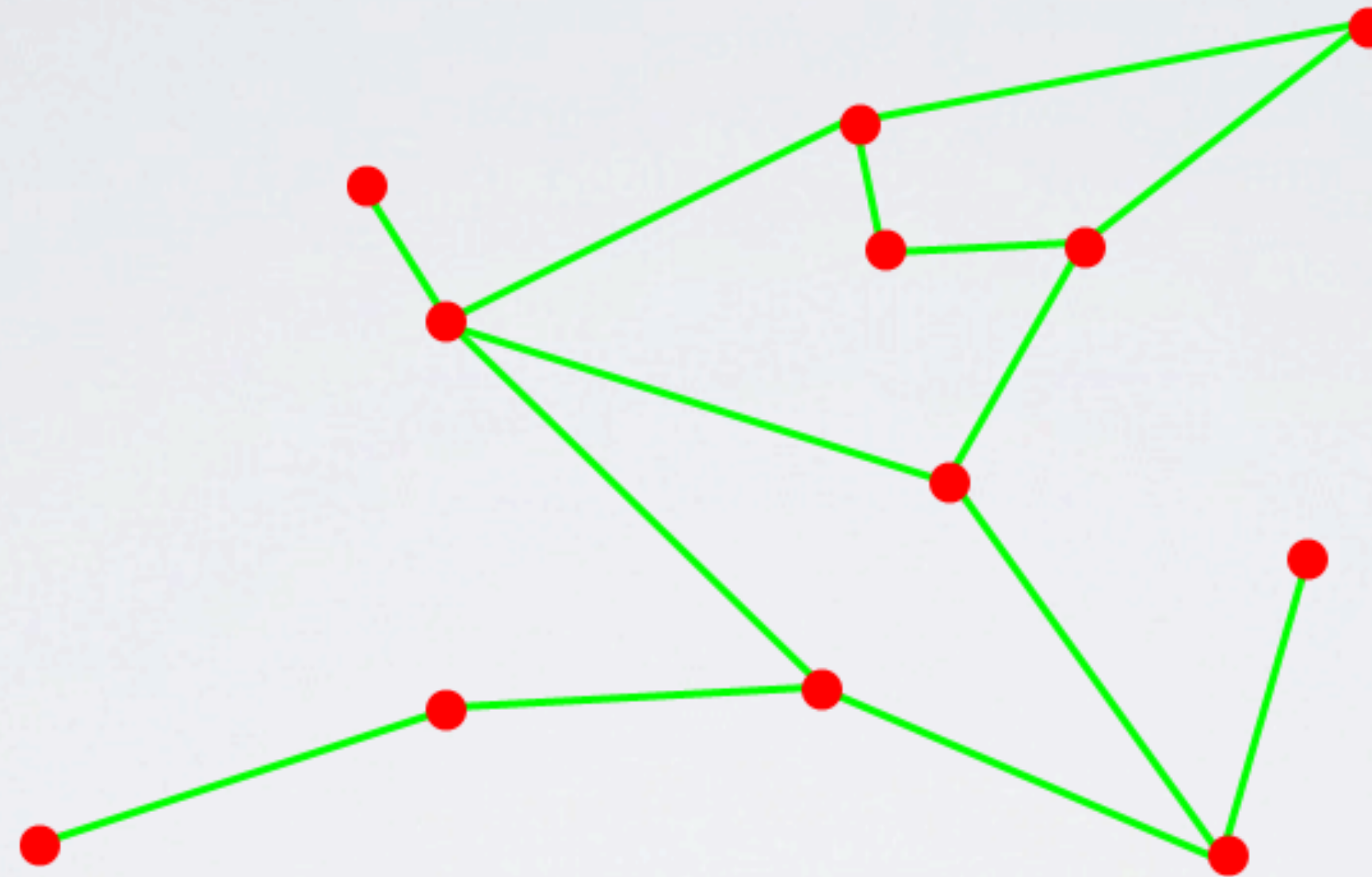
ΚΥΚΛΟΣ



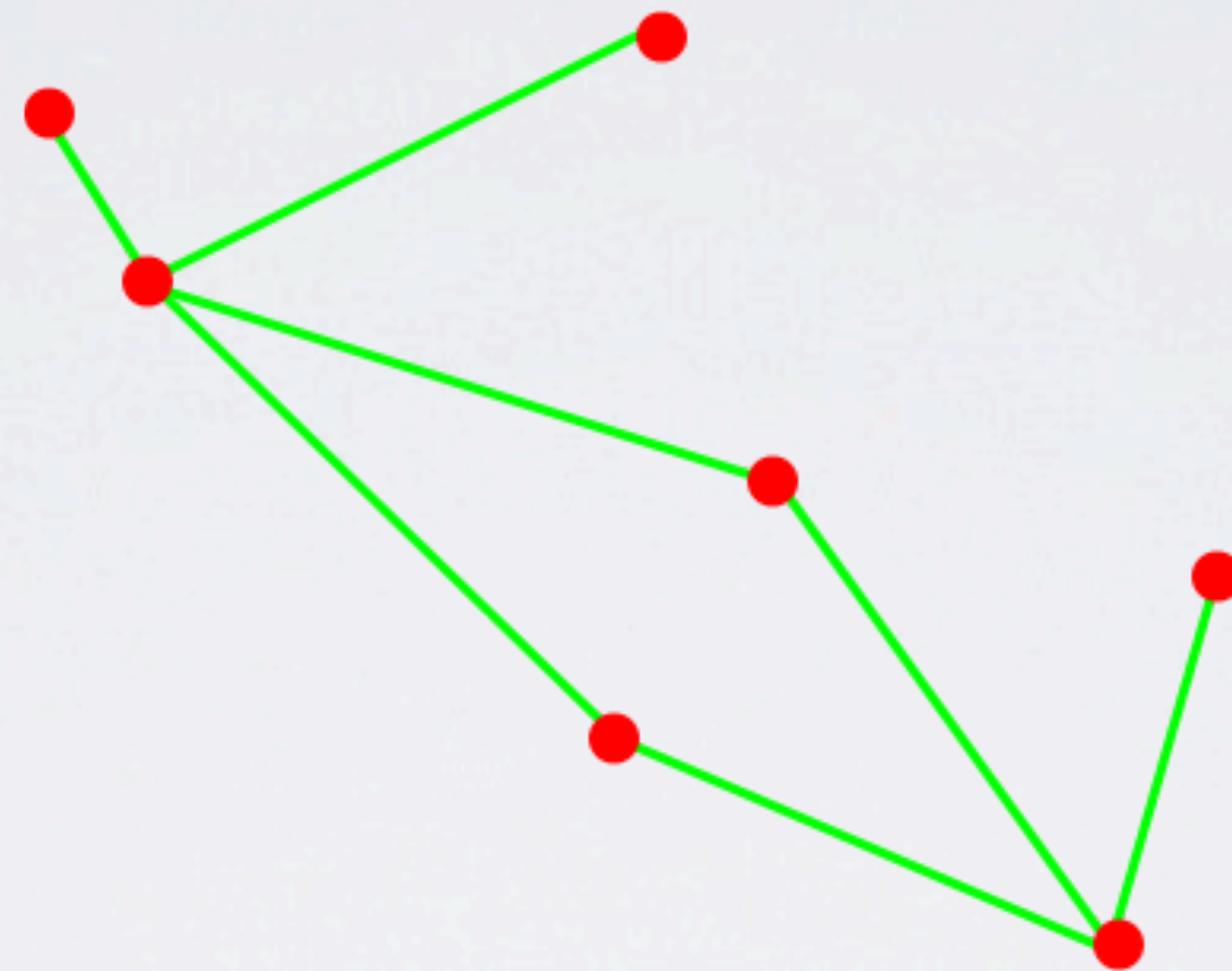


ΔΕΝΤΡΟ

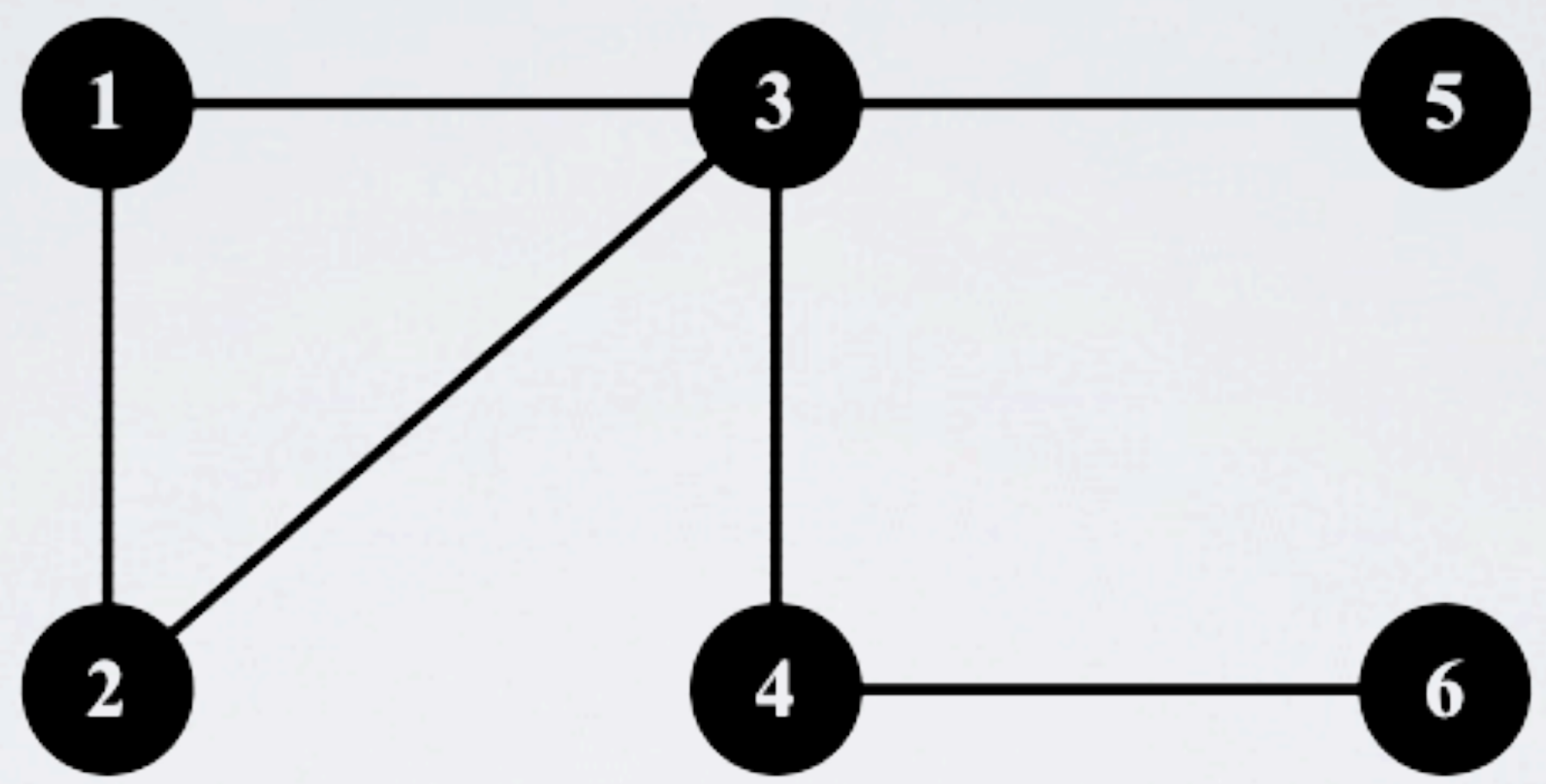
Συνεκτικός Ακυκλικός Γράφος



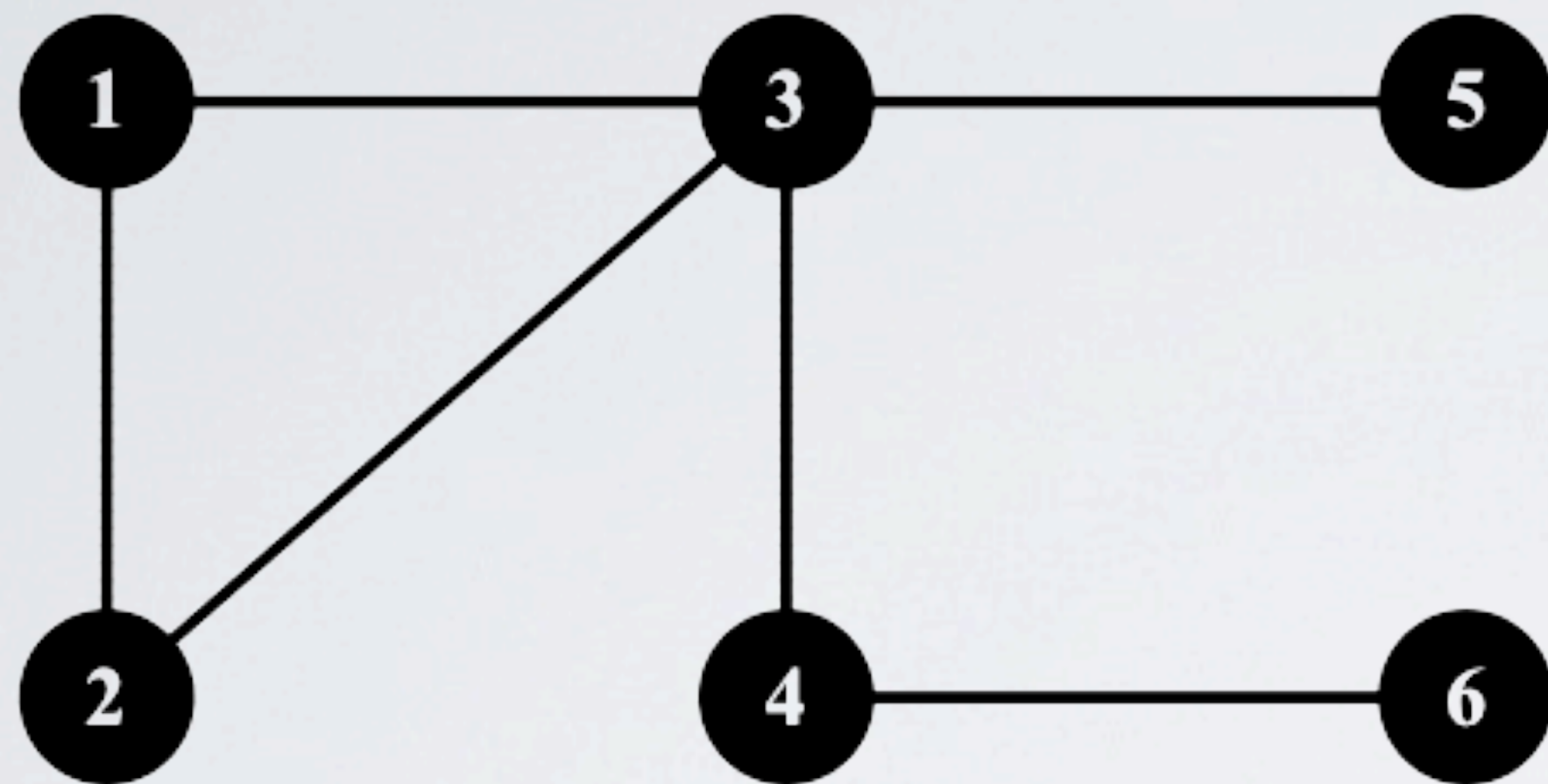
ΥΠΟΓΡΑΦΗΜΑ



ΥΠΟΓΡΑΦΗΜΑ



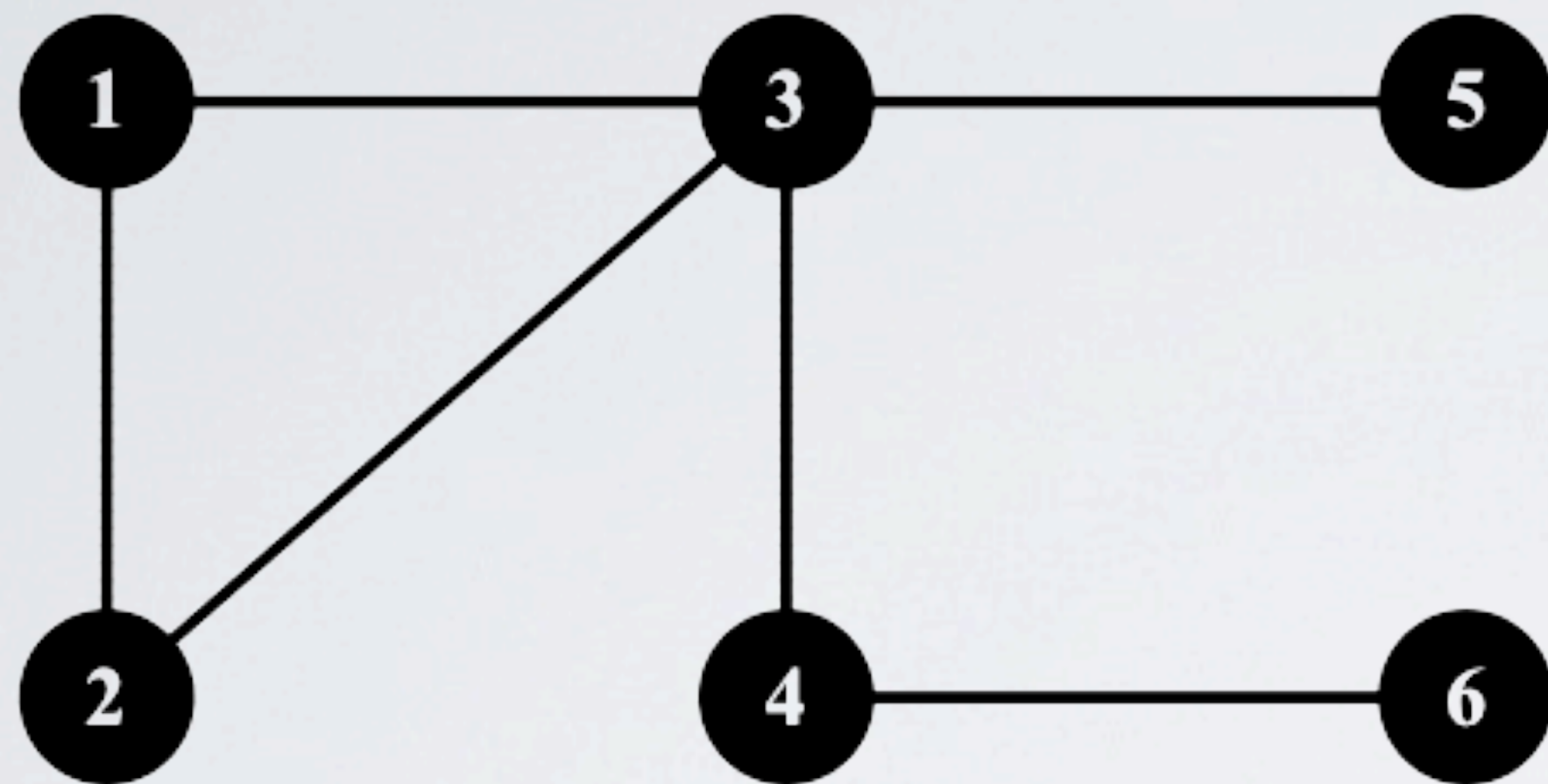
ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΡΑΦΩΝ



	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

ΠΙΝΑΚΑΣ ΓΕΙΤΝΙΑΣΗΣ

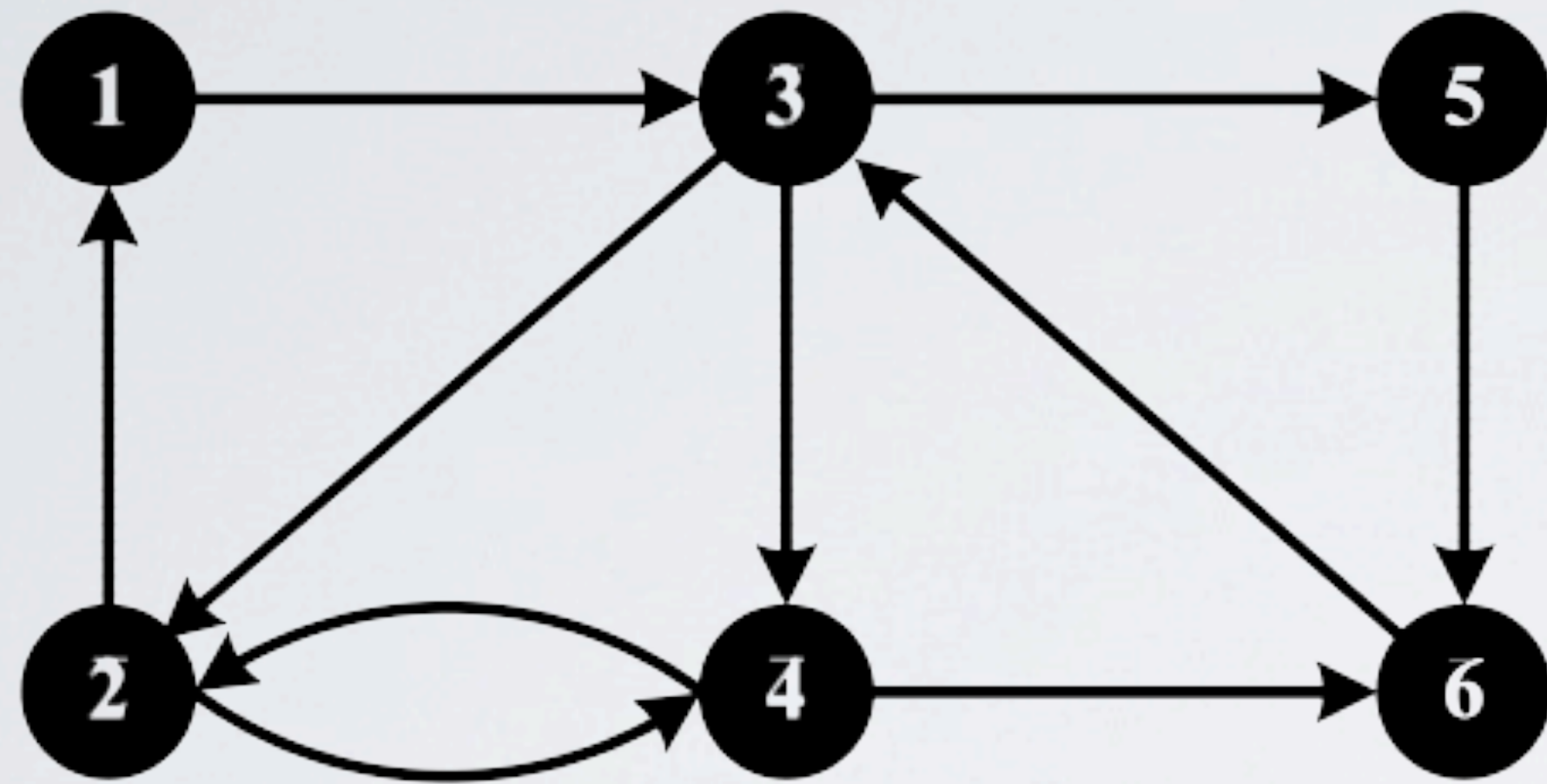
Μη κατευθυνόμενο γράφημα: Συμμετρικός Πίνακας



	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	1	0	0	0
3	1	1	0	1	1	0
4	0	0	1	0	0	1
5	0	0	1	0	0	0
6	0	0	0	1	0	0

ΠΙΝΑΚΑΣ ΓΕΙΤΝΙΑΣΗΣ

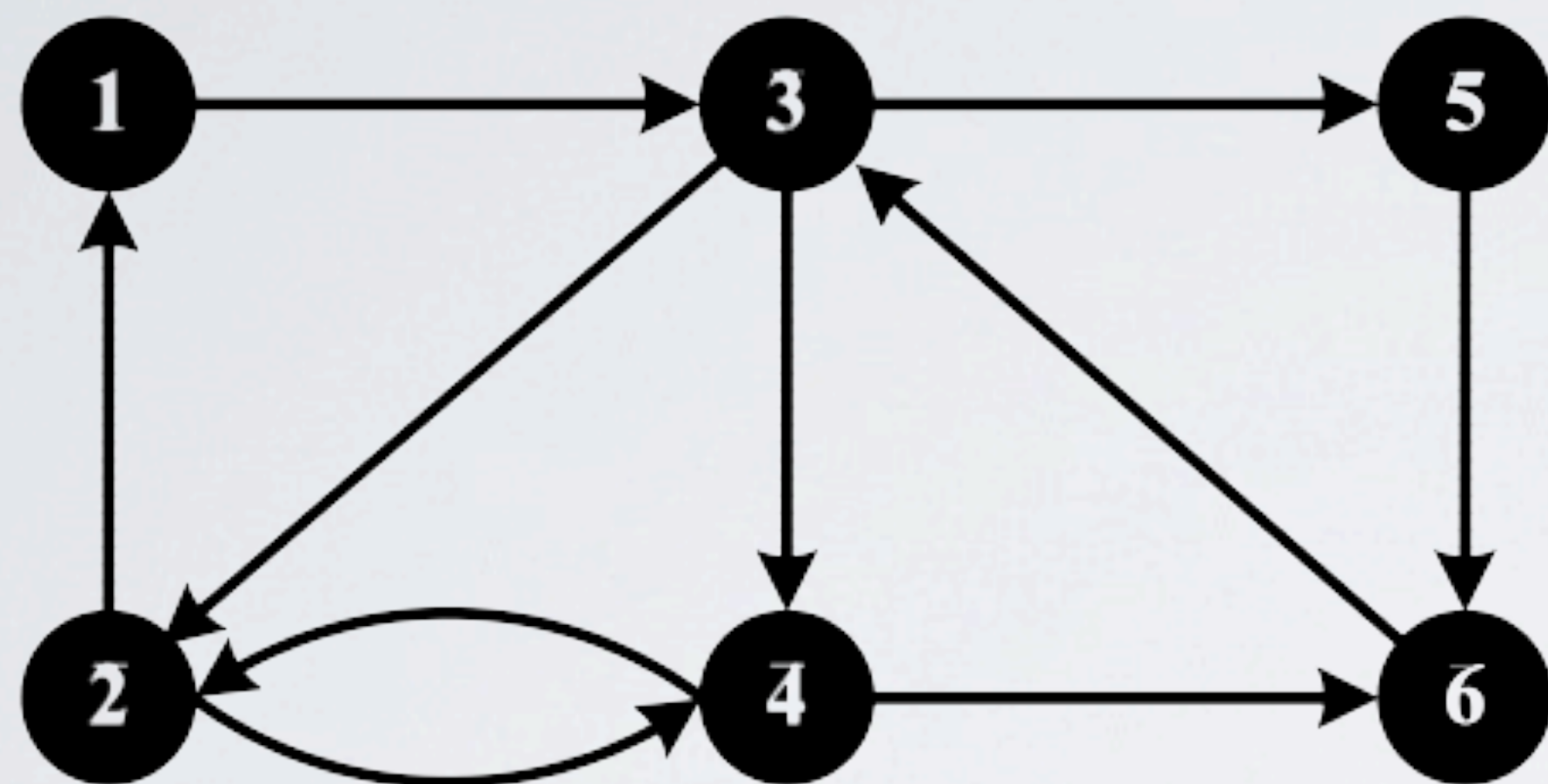
Μη κατευθυνόμενο γράφημα: Συμμετρικός Πίνακας



	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

ΠΙΝΑΚΑΣ ΓΕΙΤΝΙΑΣΗΣ

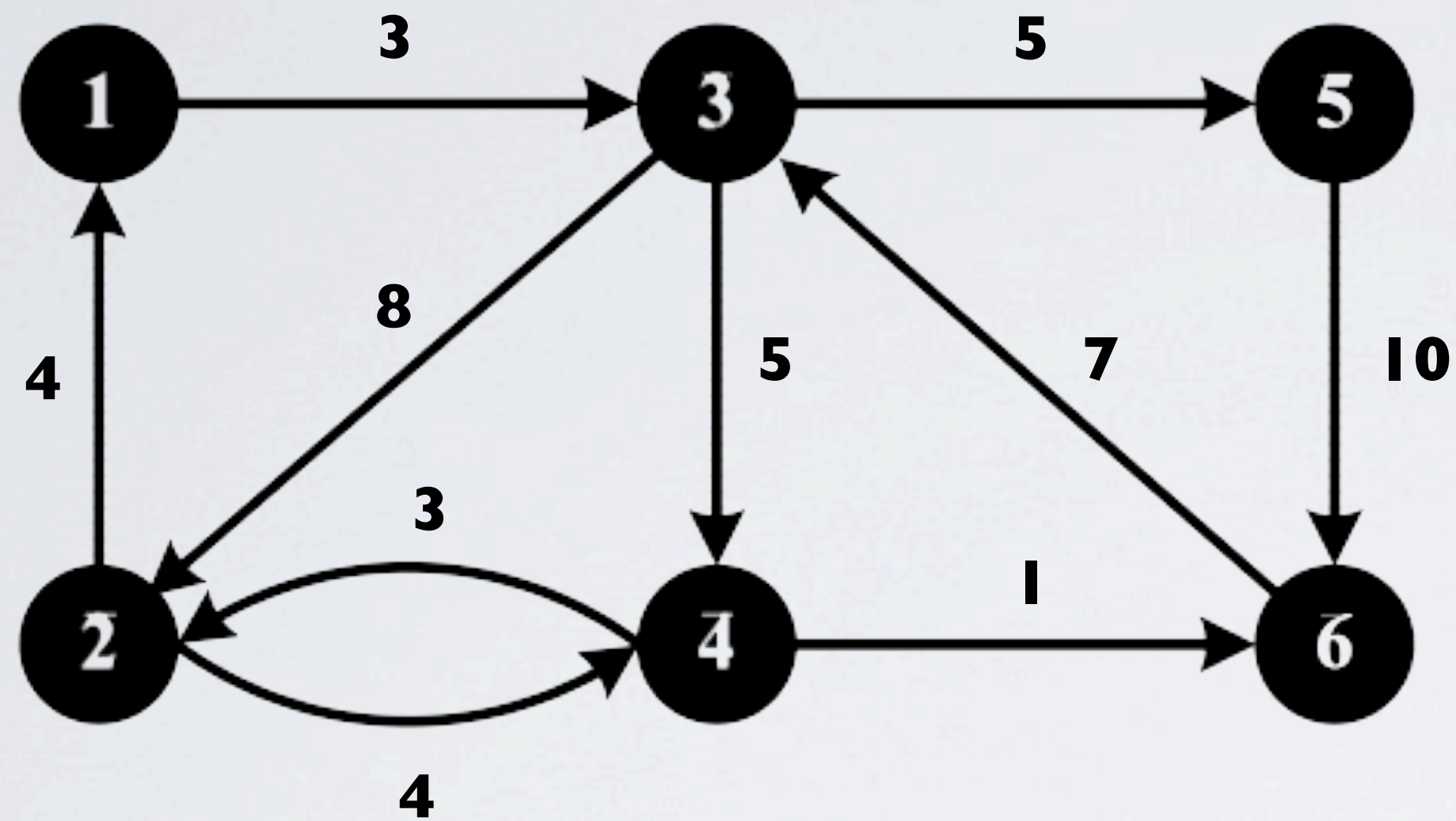
Κατευθυνόμενο γράφημα



	1	2	3	4	5	6
1	0	0	1	0	0	0
2	1	0	0	1	0	0
3	0	1	0	1	1	0
4	0	1	0	0	0	1
5	0	0	0	0	0	1
6	0	0	1	0	0	0

ΠΙΝΑΚΑΣ ΓΕΙΤΝΙΑΣΗΣ

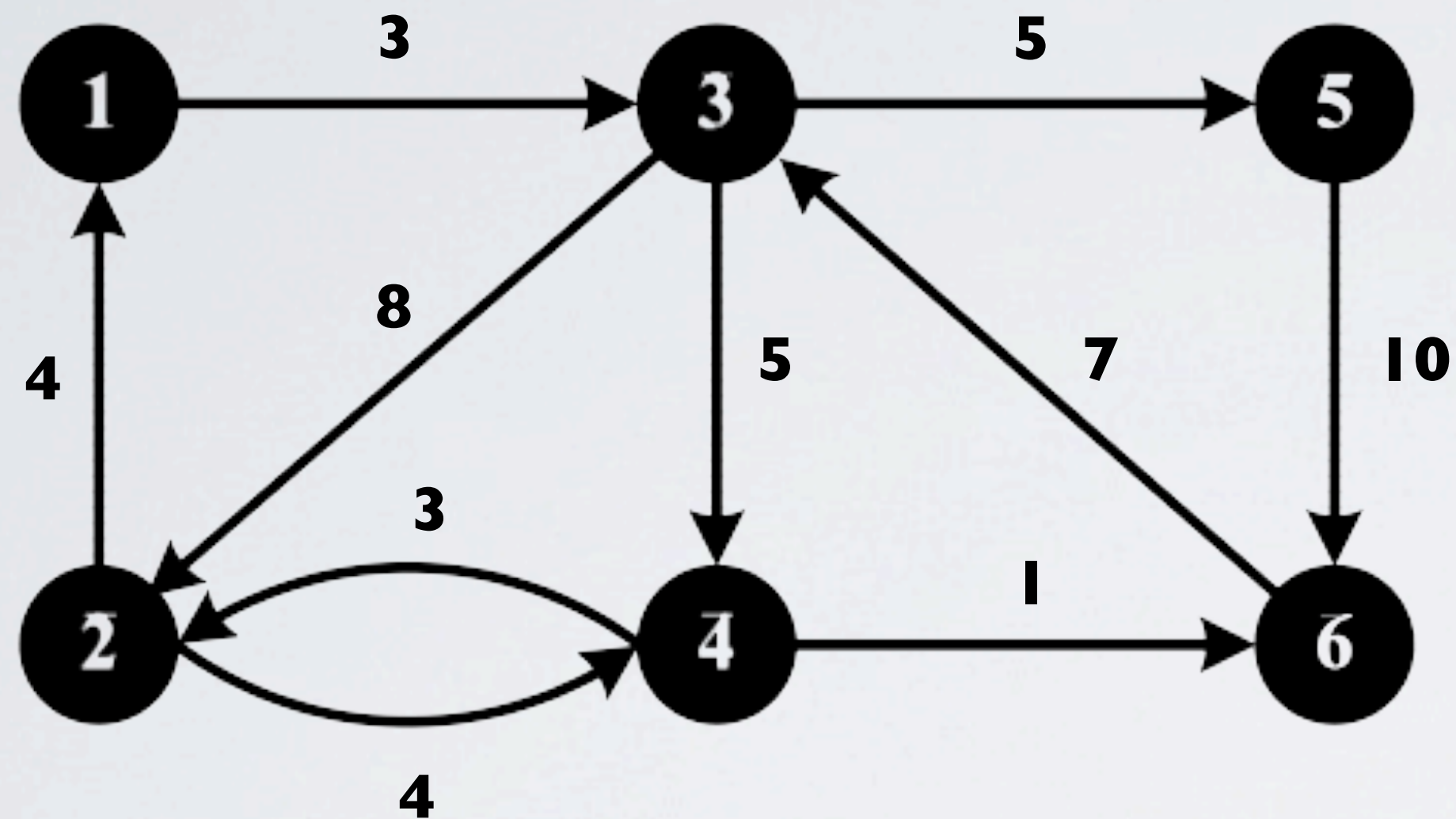
Κατευθυνόμενο γράφημα



	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

ΠΙΝΑΚΑΣ ΓΕΙΤΝΙΑΣΗΣ

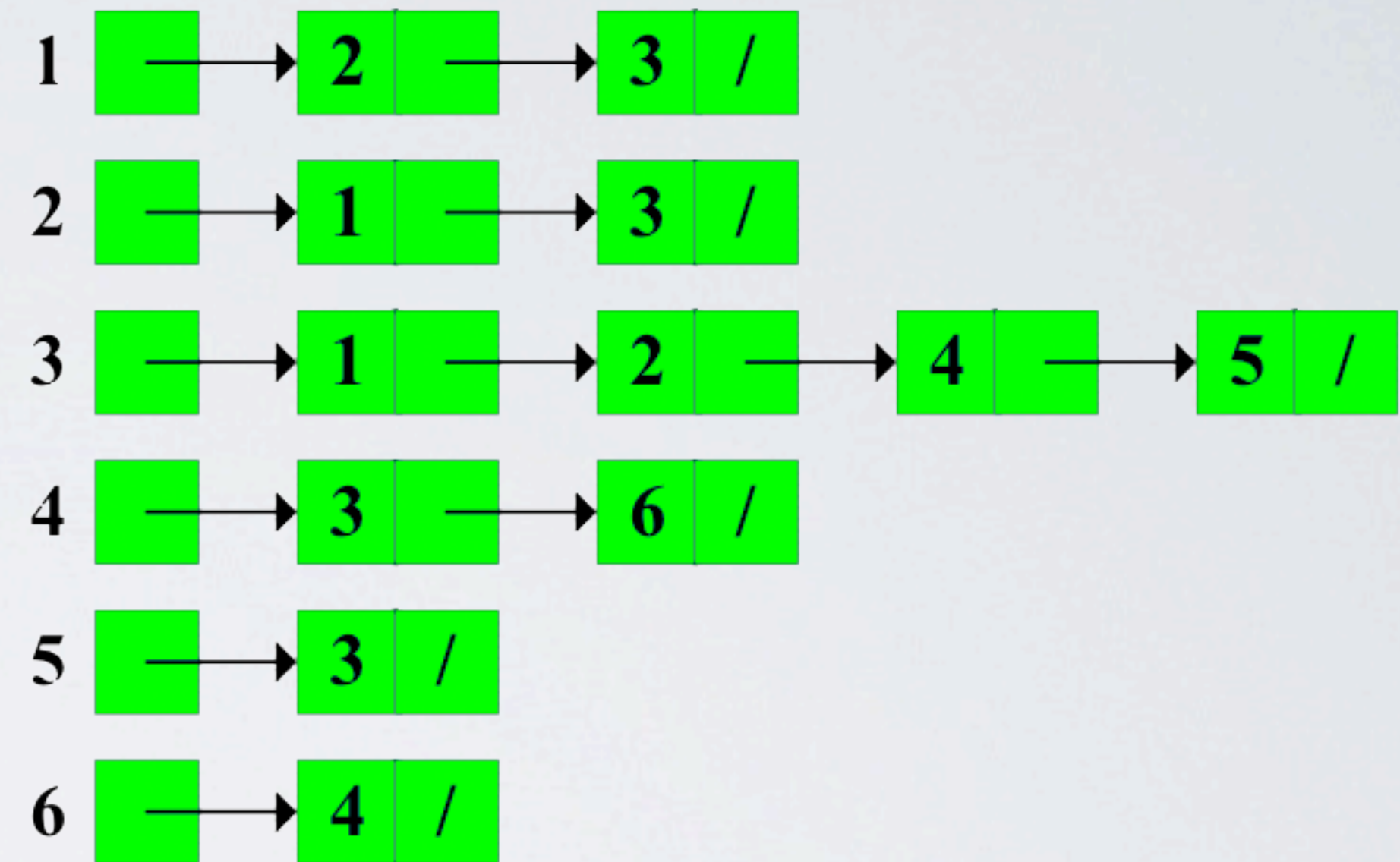
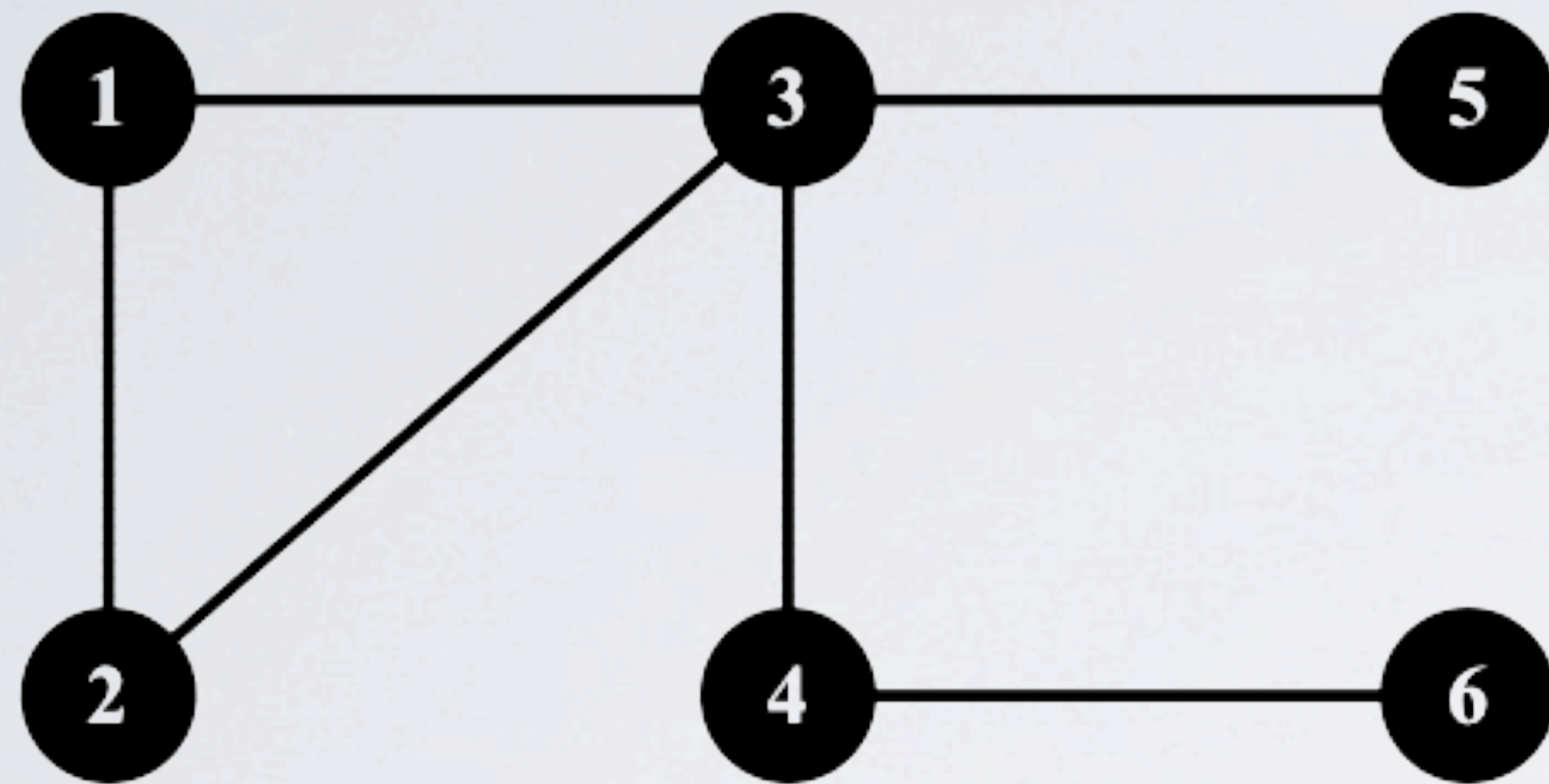
Κατευθυνόμενο γράφημα με βαρη



	1	2	3	4	5	6
1	∞	∞	3	∞	∞	∞
2	4	∞	∞	4	∞	∞
3	∞	8	∞	5	5	∞
4	∞	3	∞	∞	∞	1
5	∞	∞	∞	∞	∞	10
6	∞	∞	7	∞	∞	∞

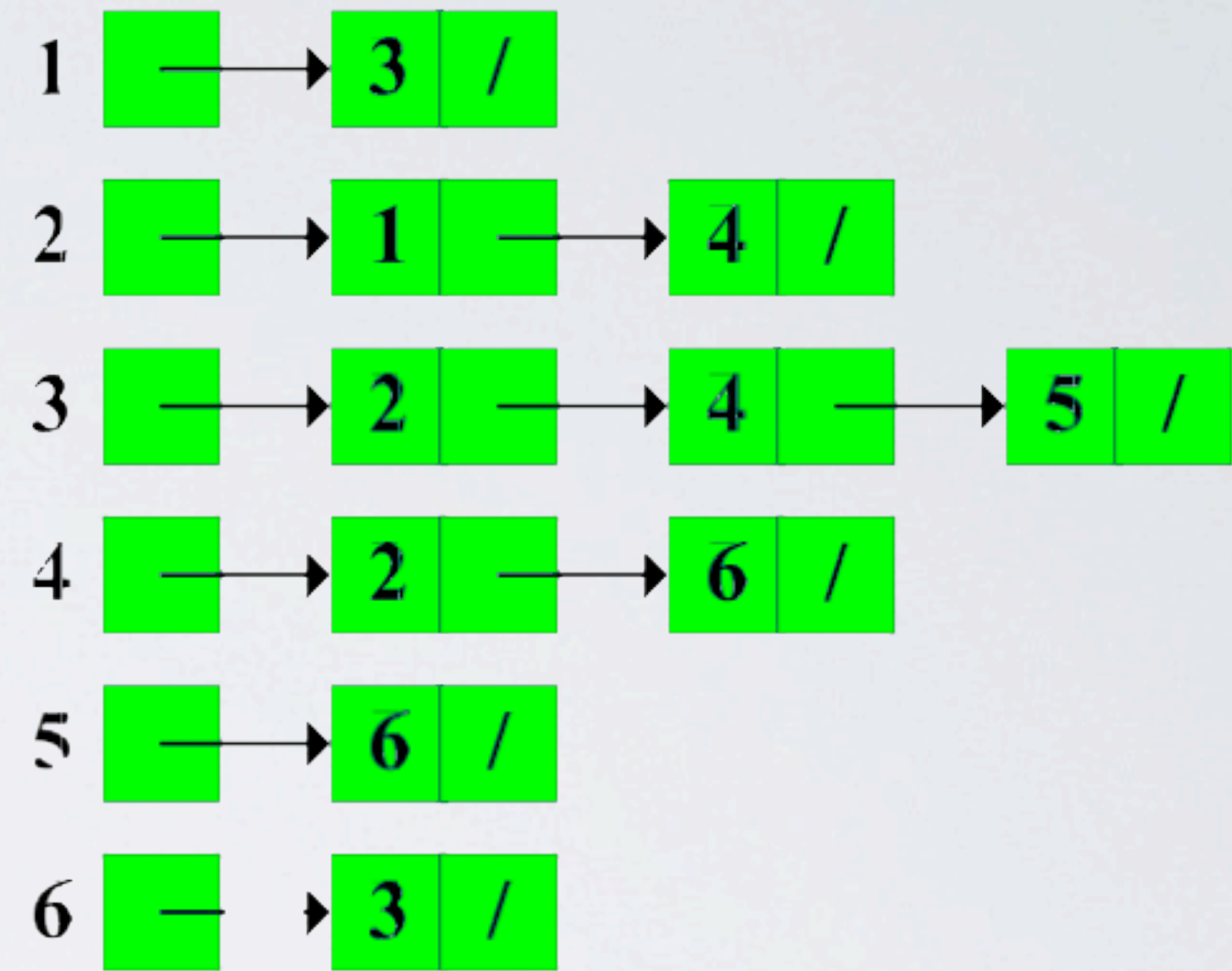
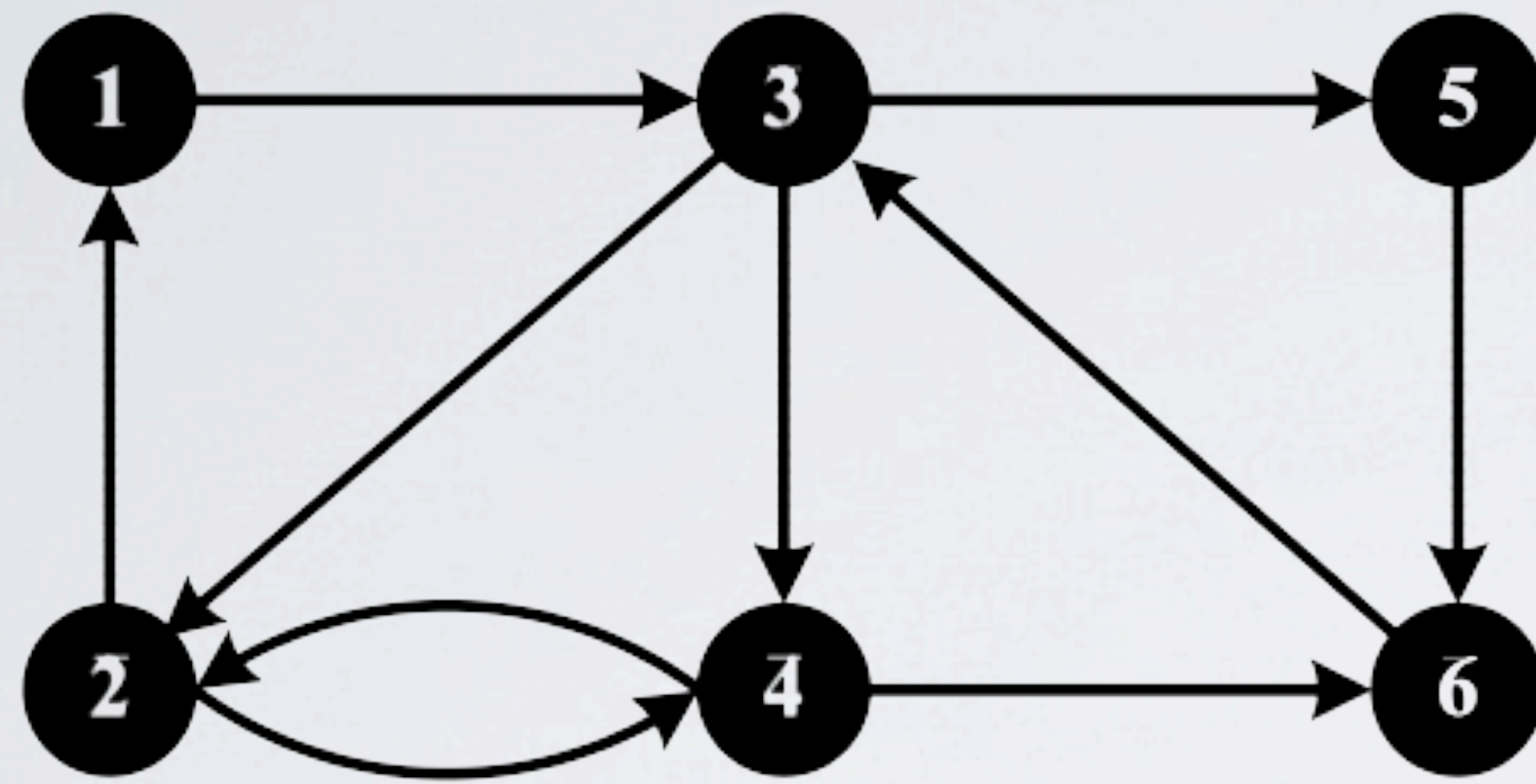
ΠΙΝΑΚΑΣ ΓΕΙΤΝΙΑΣΗΣ

Κατευθυνόμενο γράφημα με βαρη



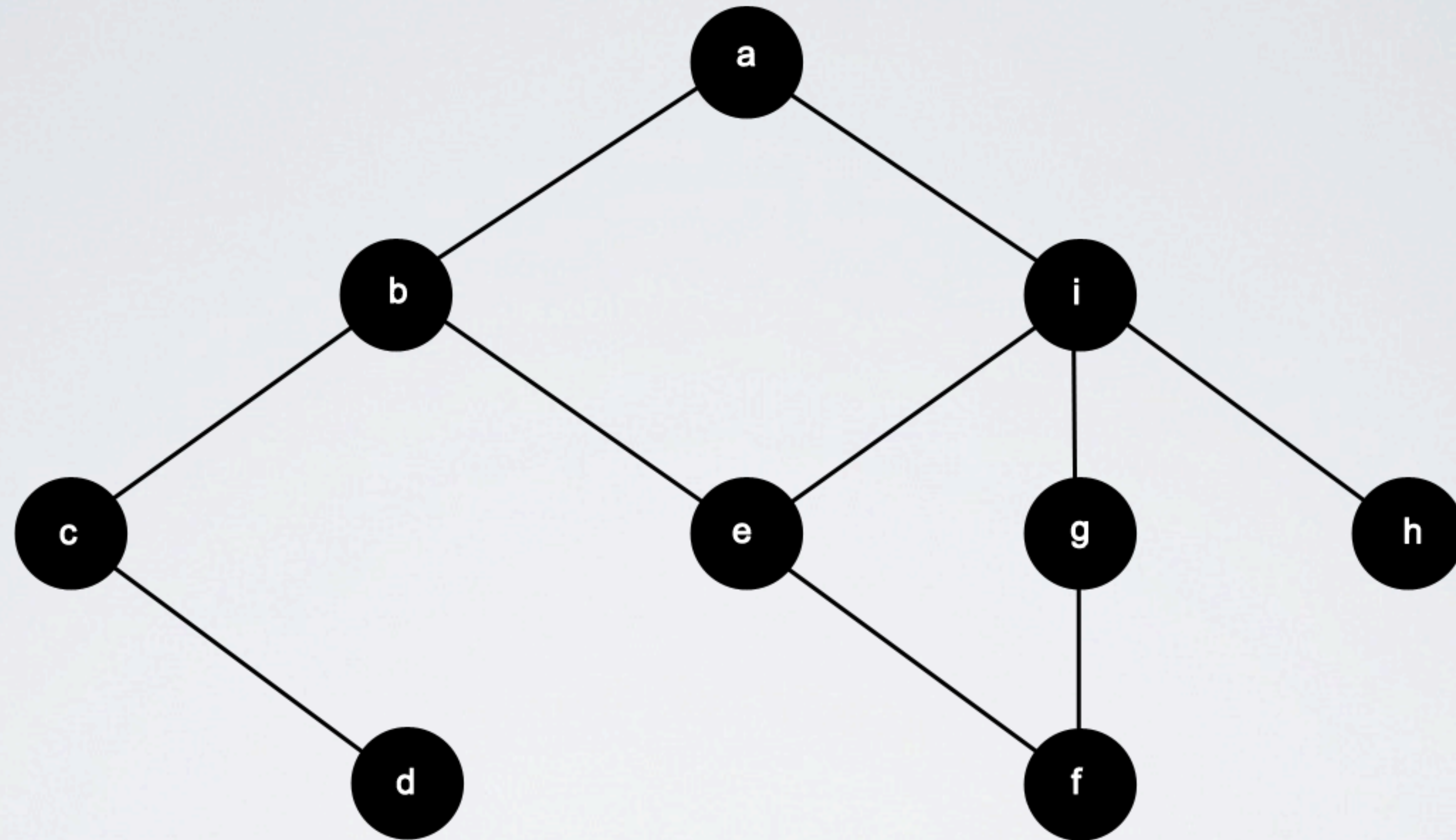
ΛΙΣΤΑ ΓΕΙΤΝΙΑΣΗΣ

Μη κατευθυνόμενο γράφημα: Συμμετρικός Πίνακας



ΛΙΣΤΑ ΓΕΙΤΝΙΑΣΗΣ

Κατευθυνόμενο γράφημα

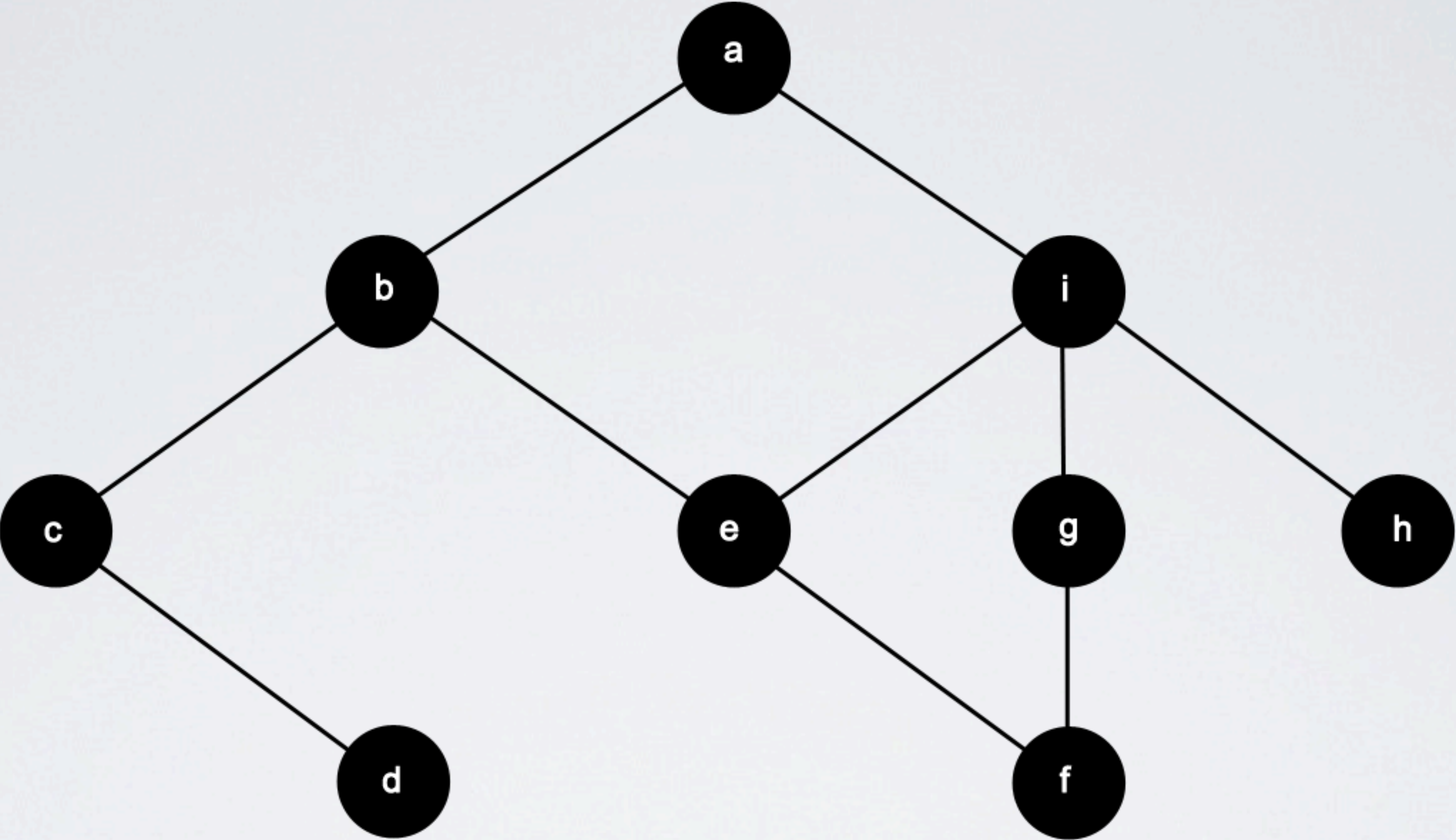


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

BFS

Queue

Node	Parent

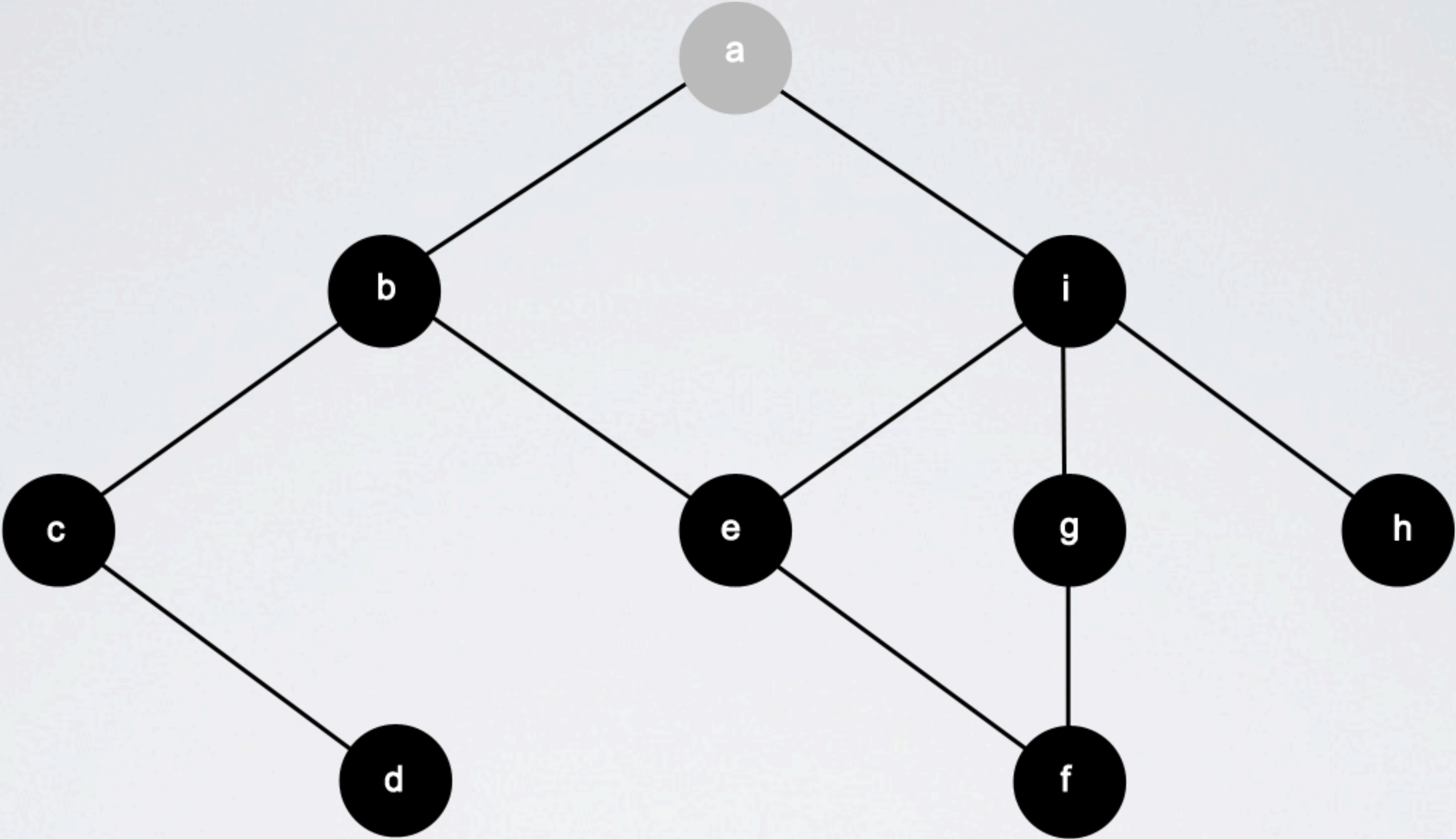


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

BFS

Queue

Node	Parent
a	-

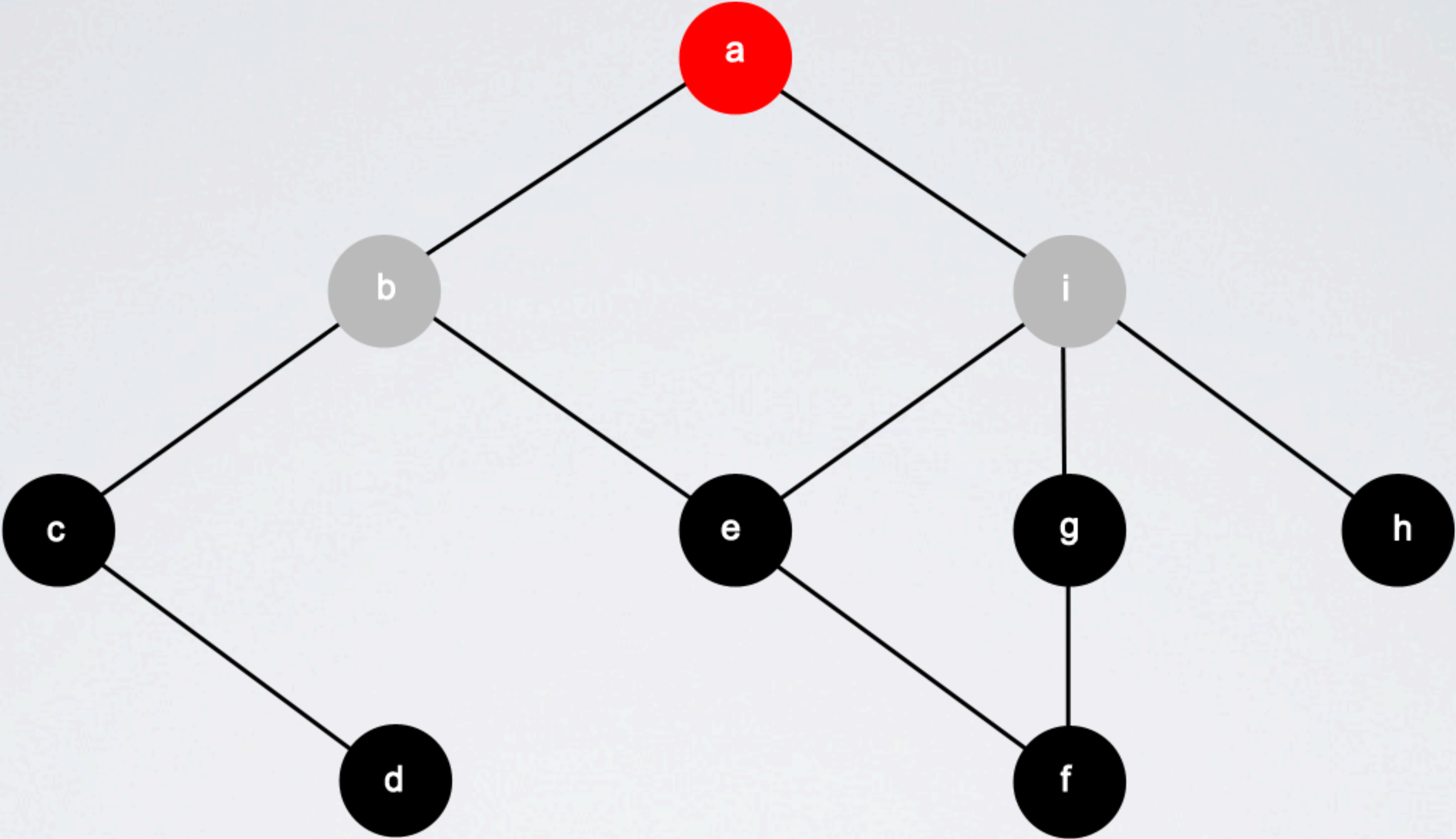


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [a]

Queue

Node	Parent
b	a
i	a

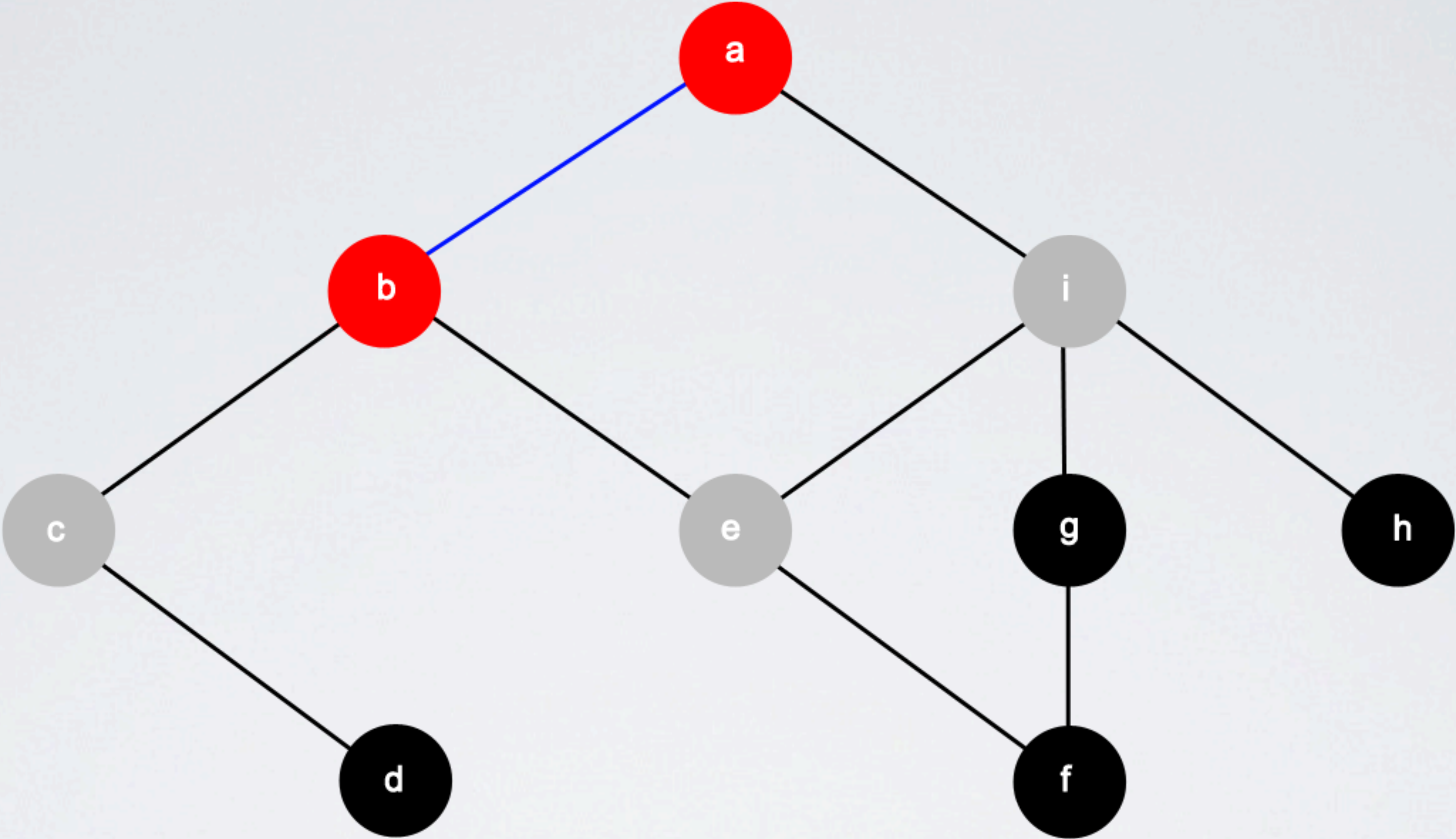


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [a, b, i]

Queue

Node	Parent
i	a
c	b
e	b

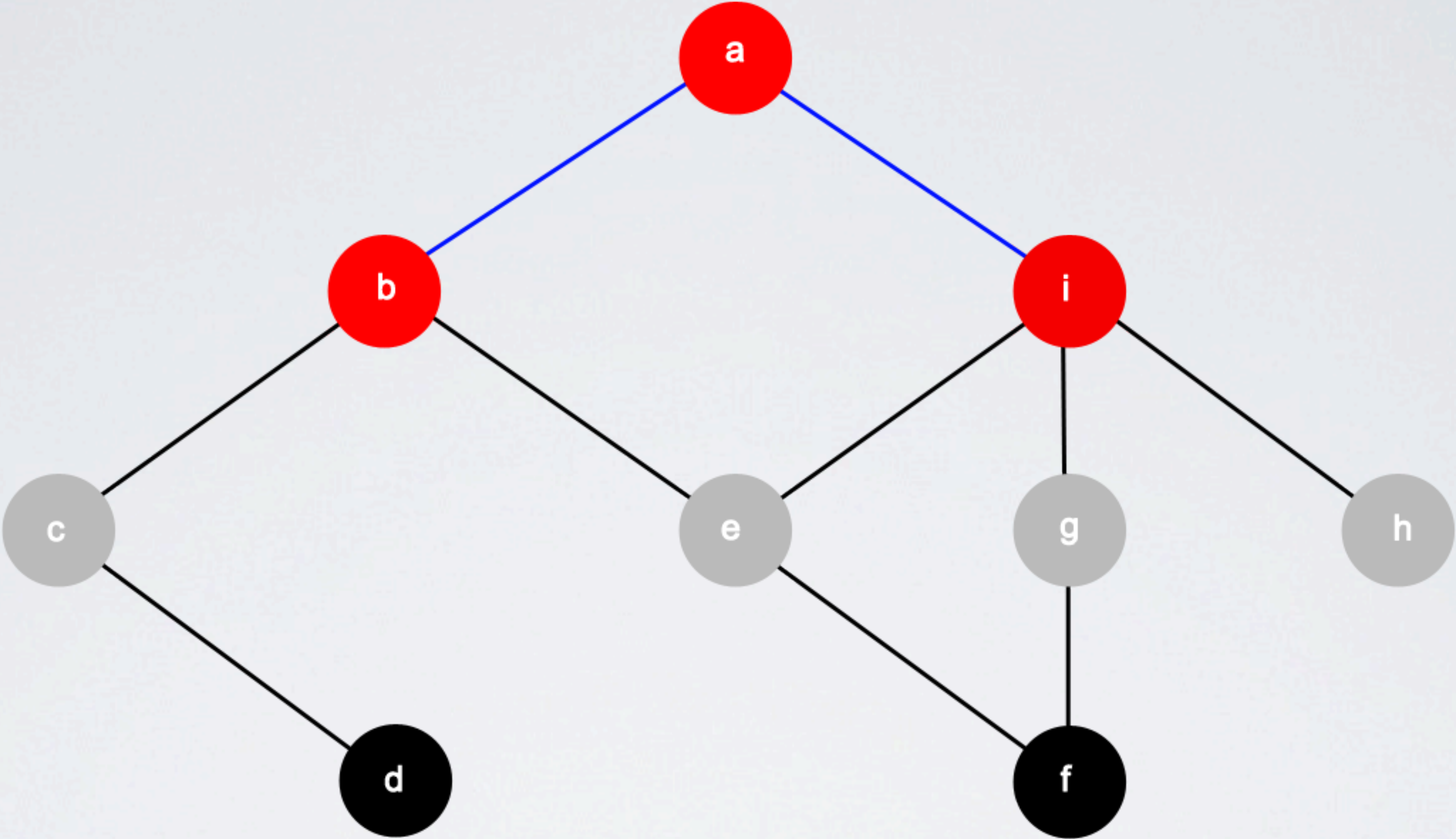


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [b, i, c, e]

Queue

Node	Parent
c	b
e	b
g	i
h	i

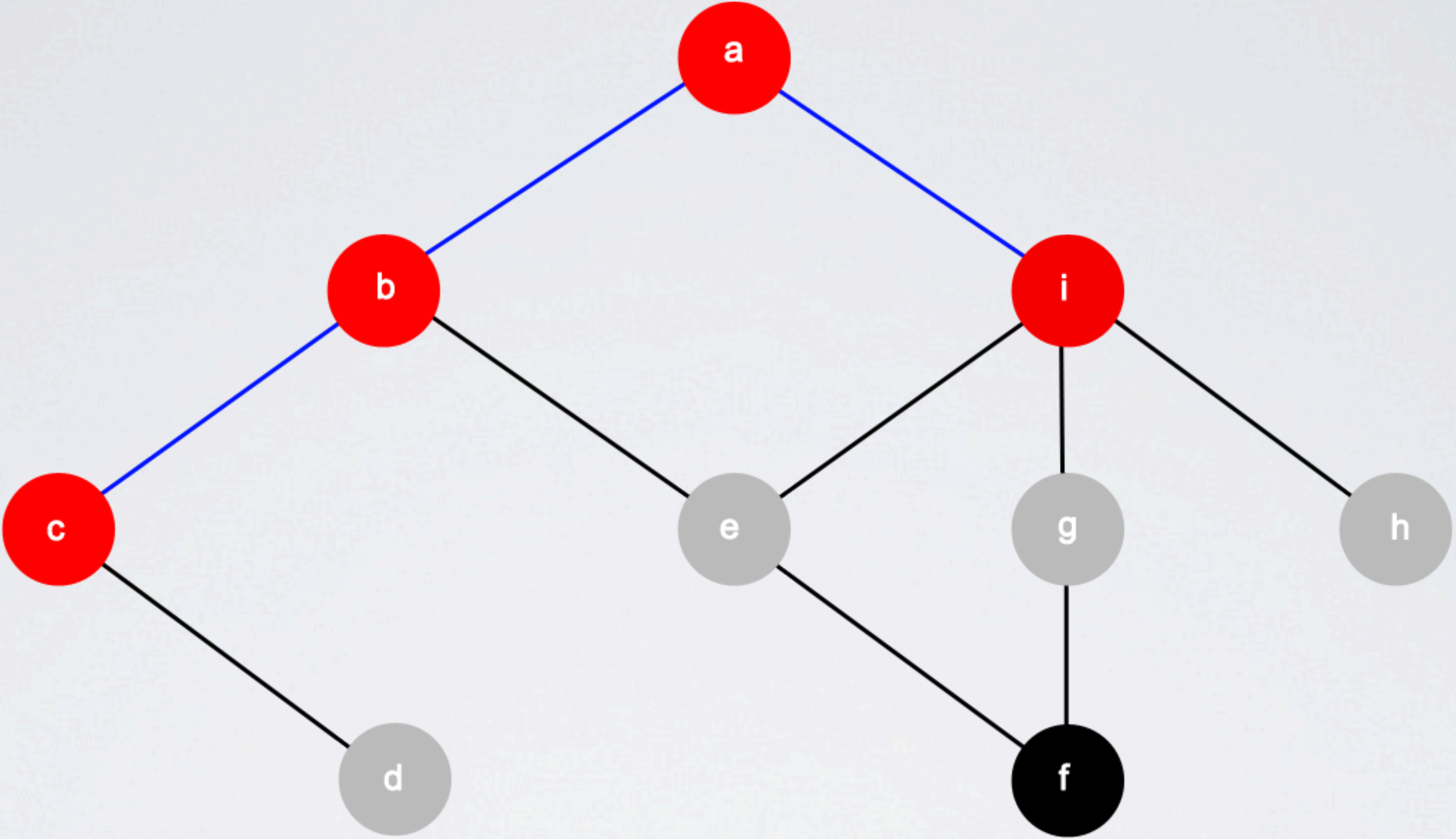


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [i, c, e, g, h]

Queue

Node	Parent
e	b
g	i
h	i
d	c

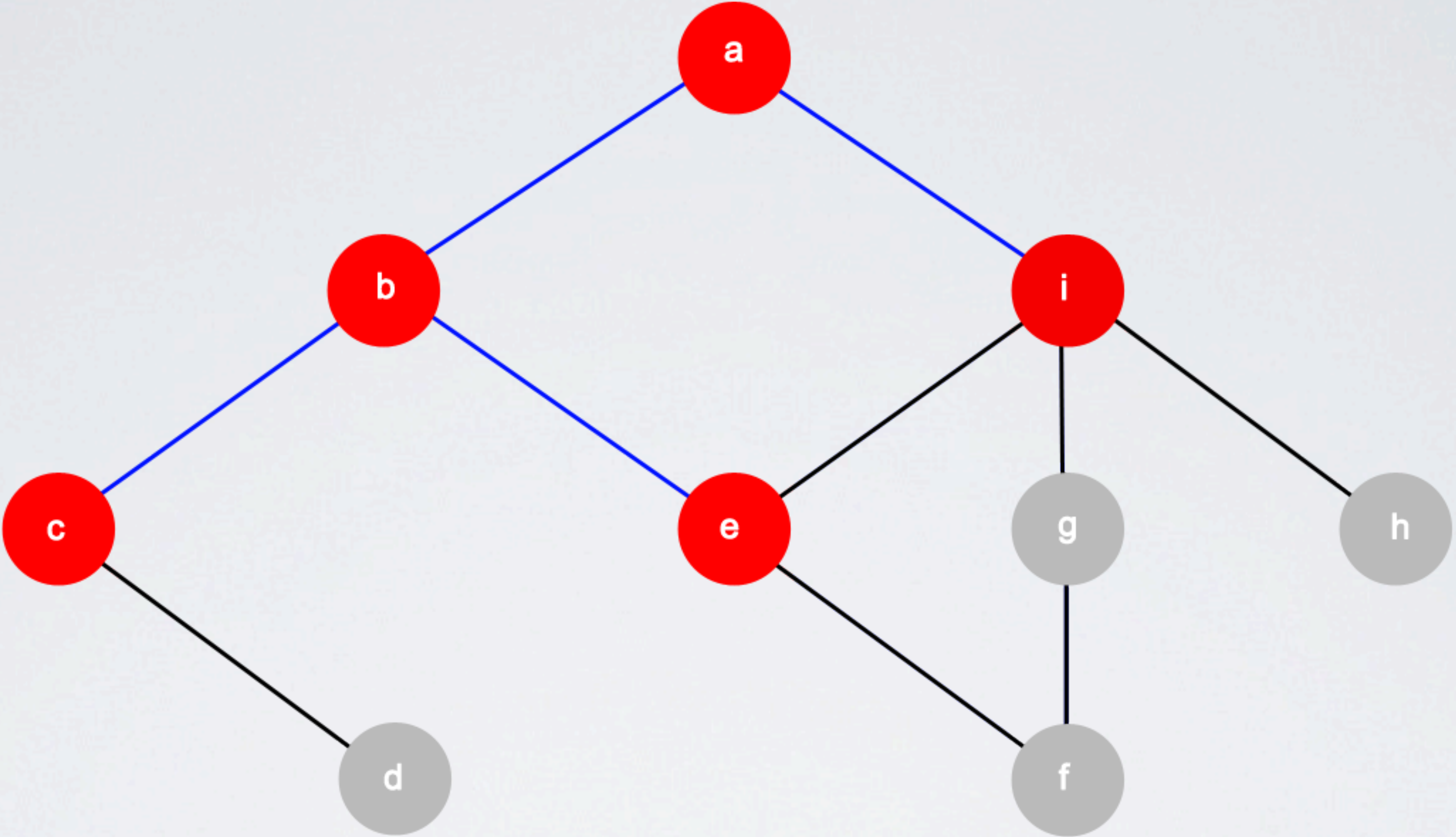


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [ε, e, g, h, d]

Queue

Node	Parent
g	i
h	i
d	c
f	e

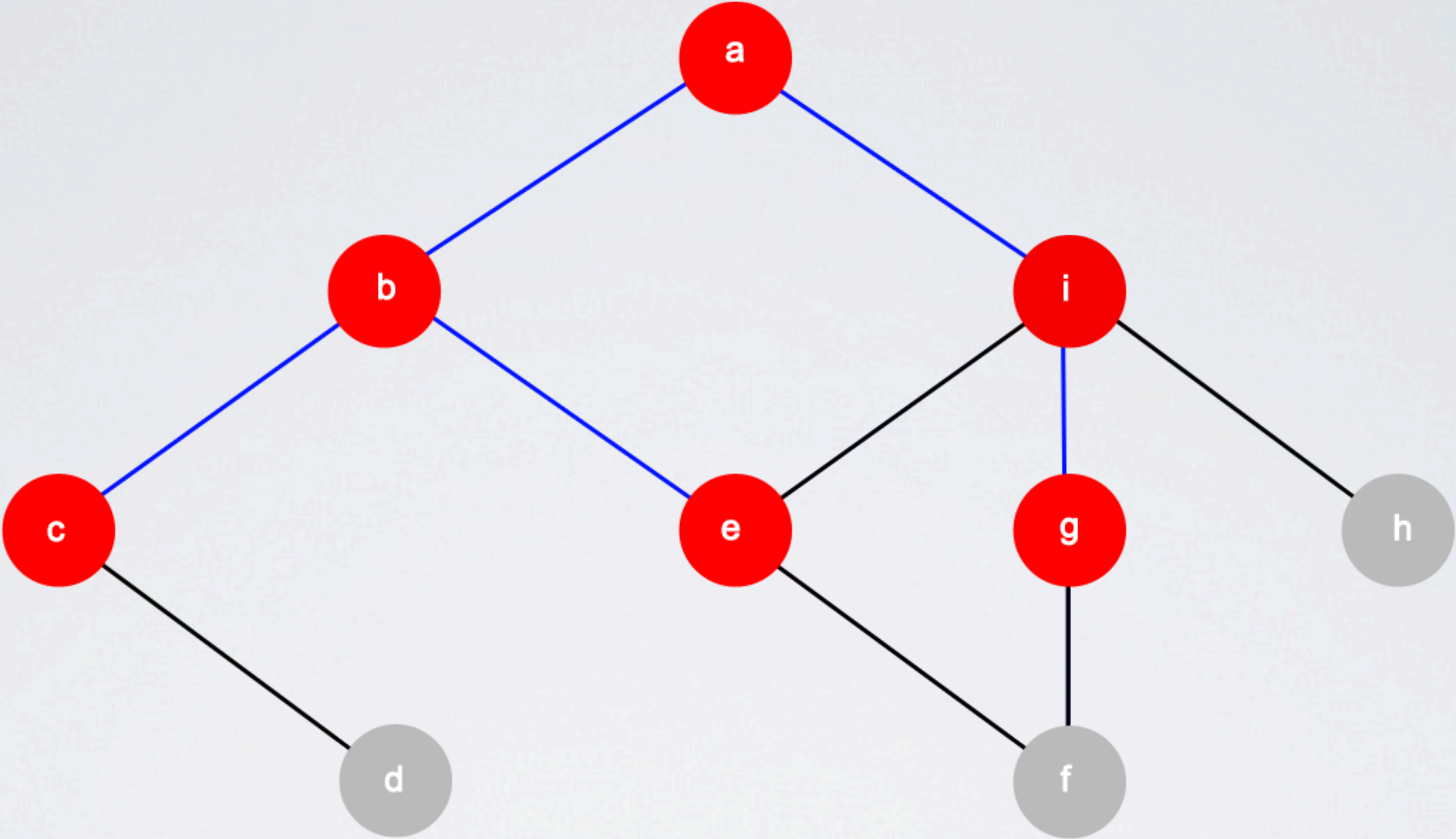


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [e, g, h, d, f]

Queue

Node	Parent
h	i
d	c
f	e

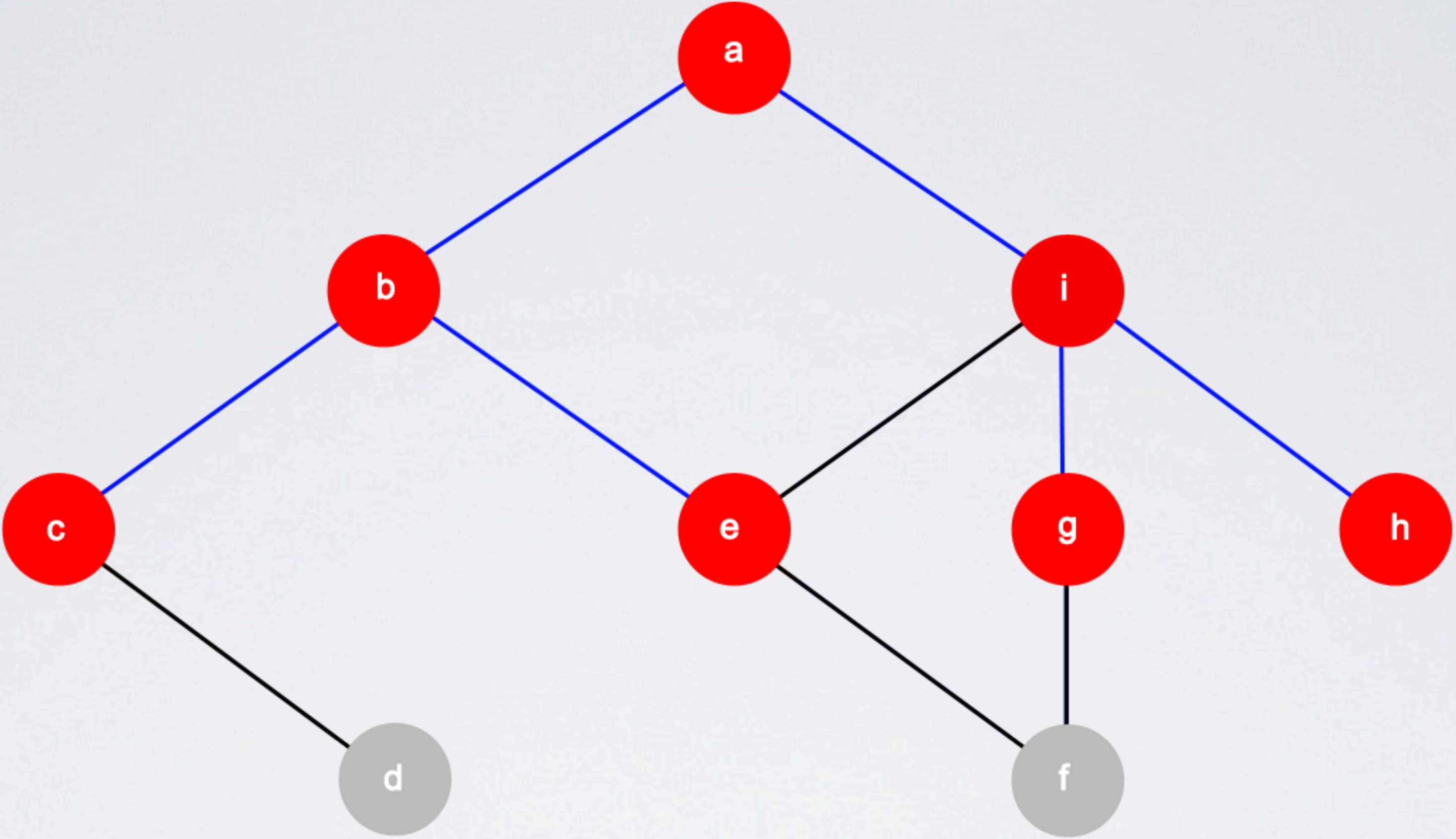


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [g, h, d, f]

Queue

Node	Parent
d	c
f	e

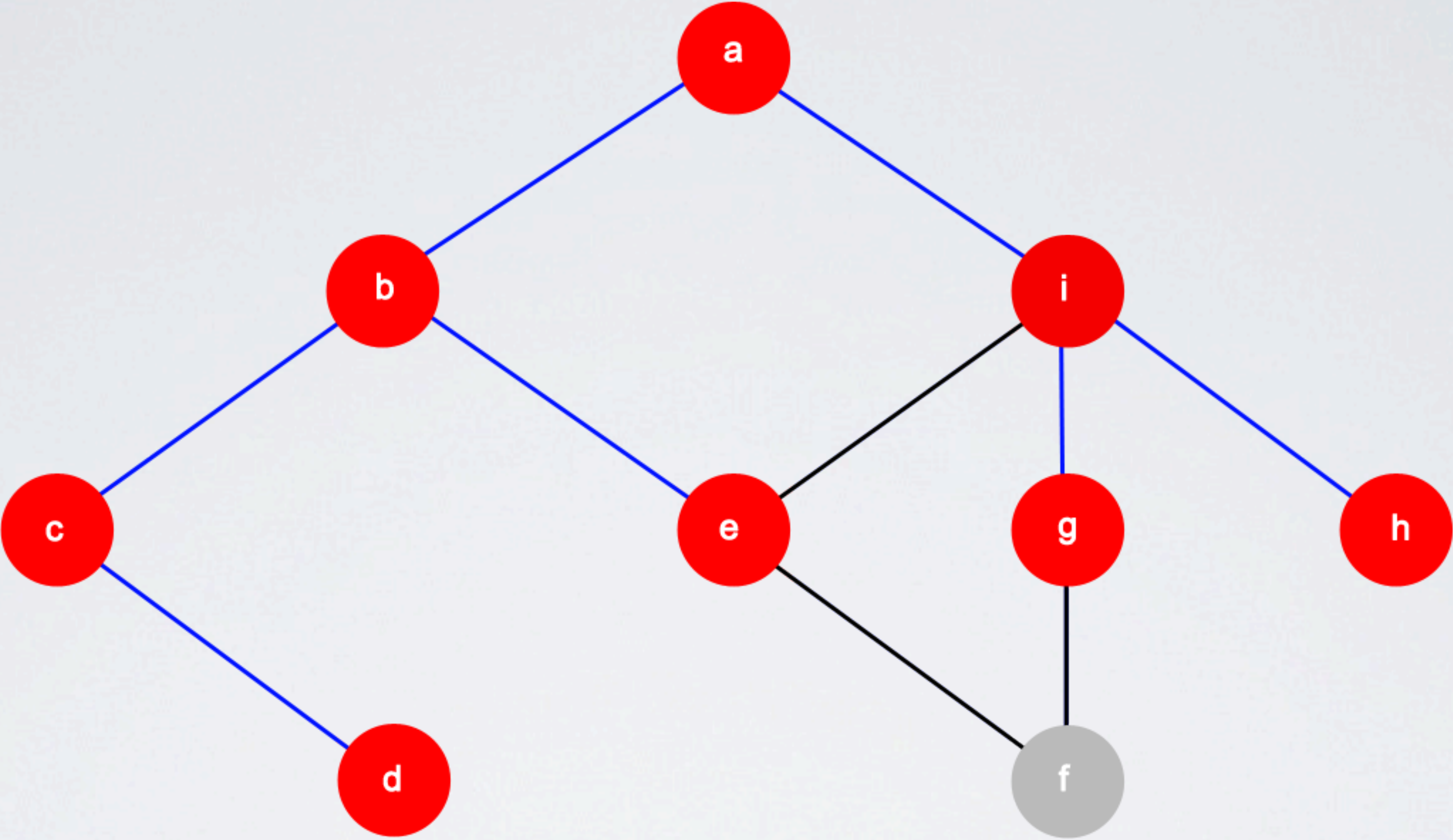


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [h, d, f]

Queue

Node	Parent
f	e

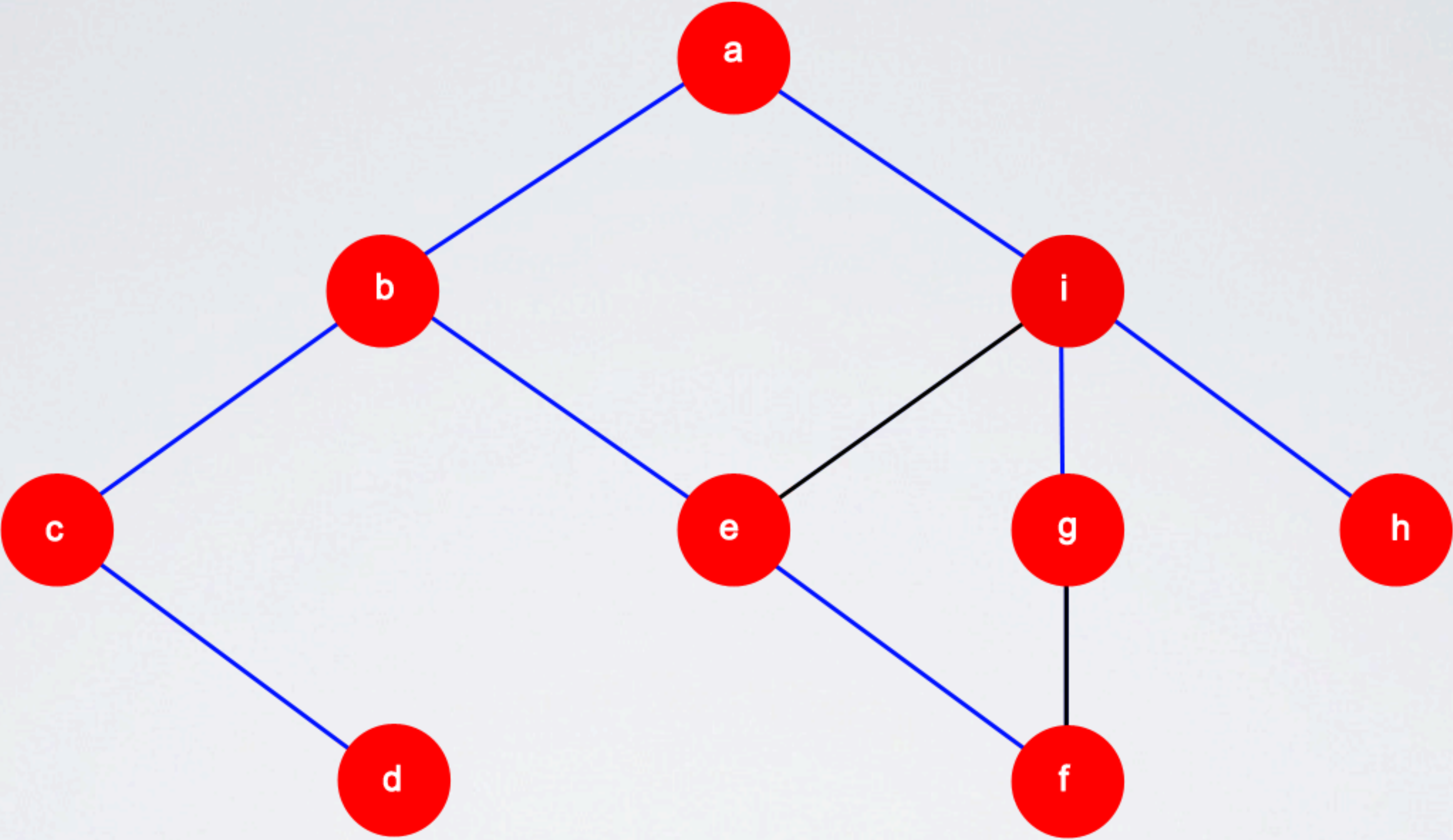


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

Queue: [d, f]

Queue

Node	Parent



ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

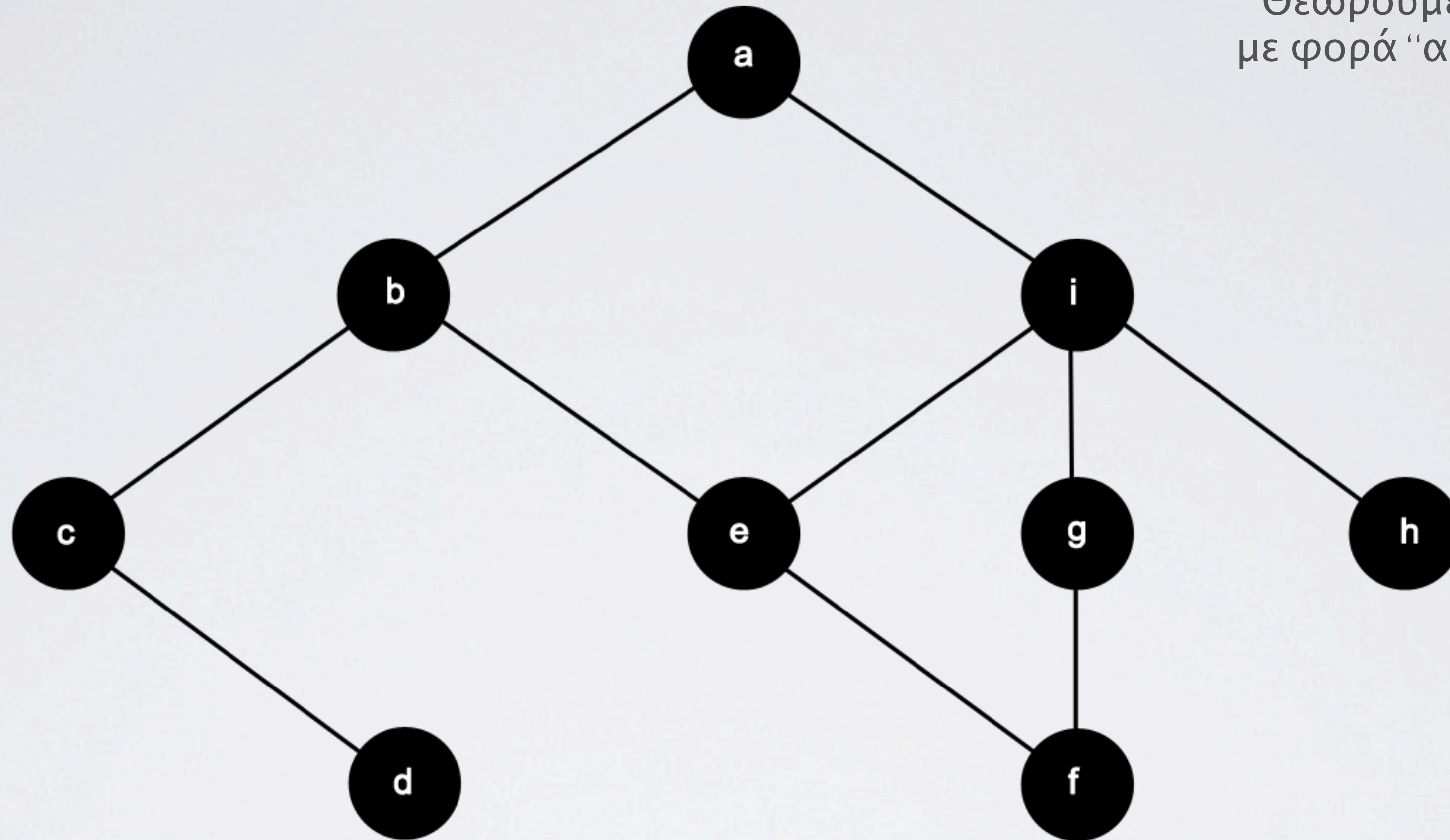
Queue: [f]

ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΠΛΑΤΟΣ

```
bfs(graph, node)
{
    create a queue q
    enqueue node onto q
    mark node
    parent[node] = null
    while q is not empty
    {
        temp = q.dequeue()
        for each edge e incident on node in graph
        {
            let w be the other end of e
```

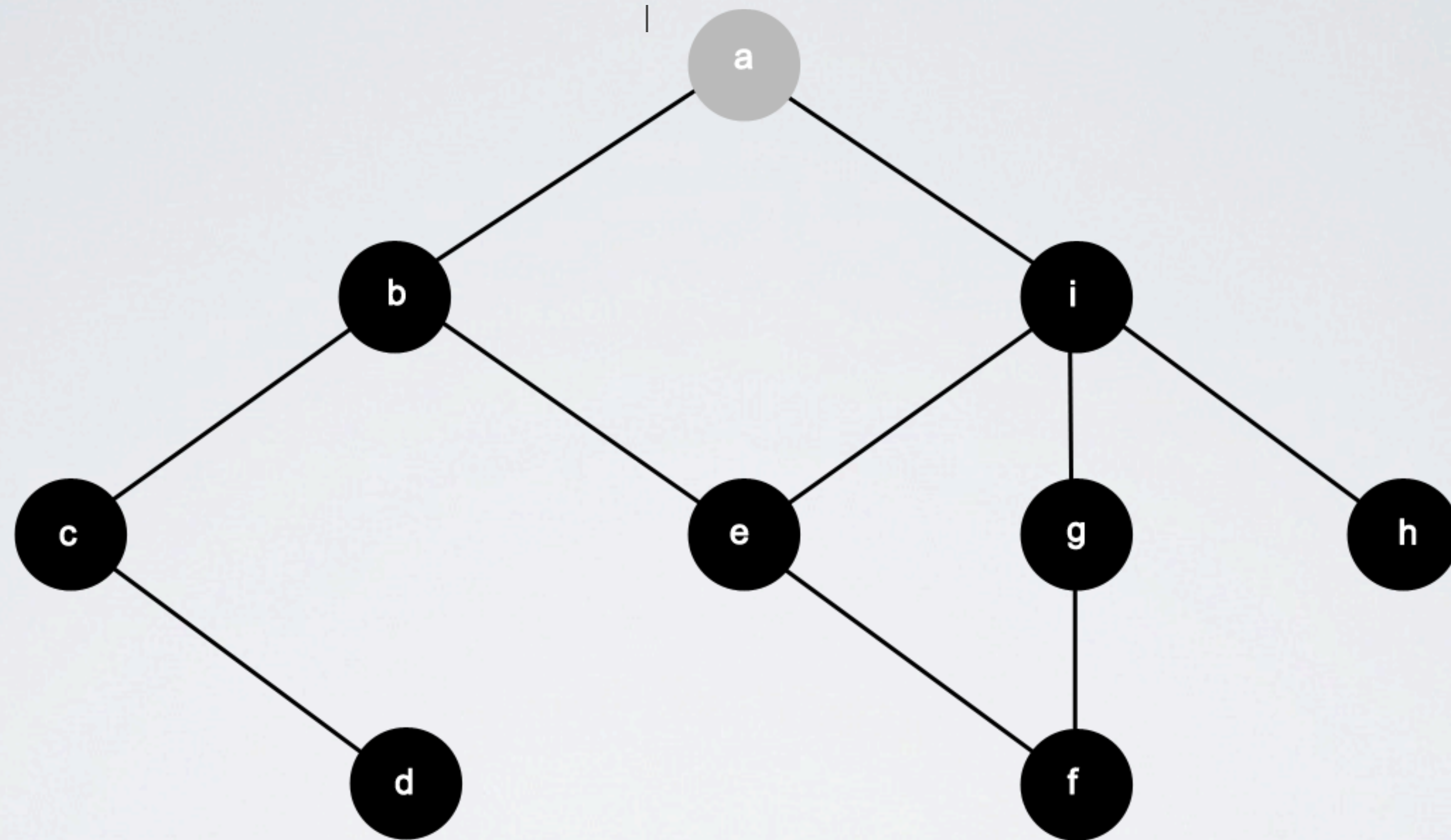
```
            if w is not marked
            {
                parent[w] = temp
                mark w
                enqueue w onto q
            }
        }
    }
}
```


Θεωρούμε κατευθυνόμενο γράφο
με φορά “από πάνω προς τα κάτω”



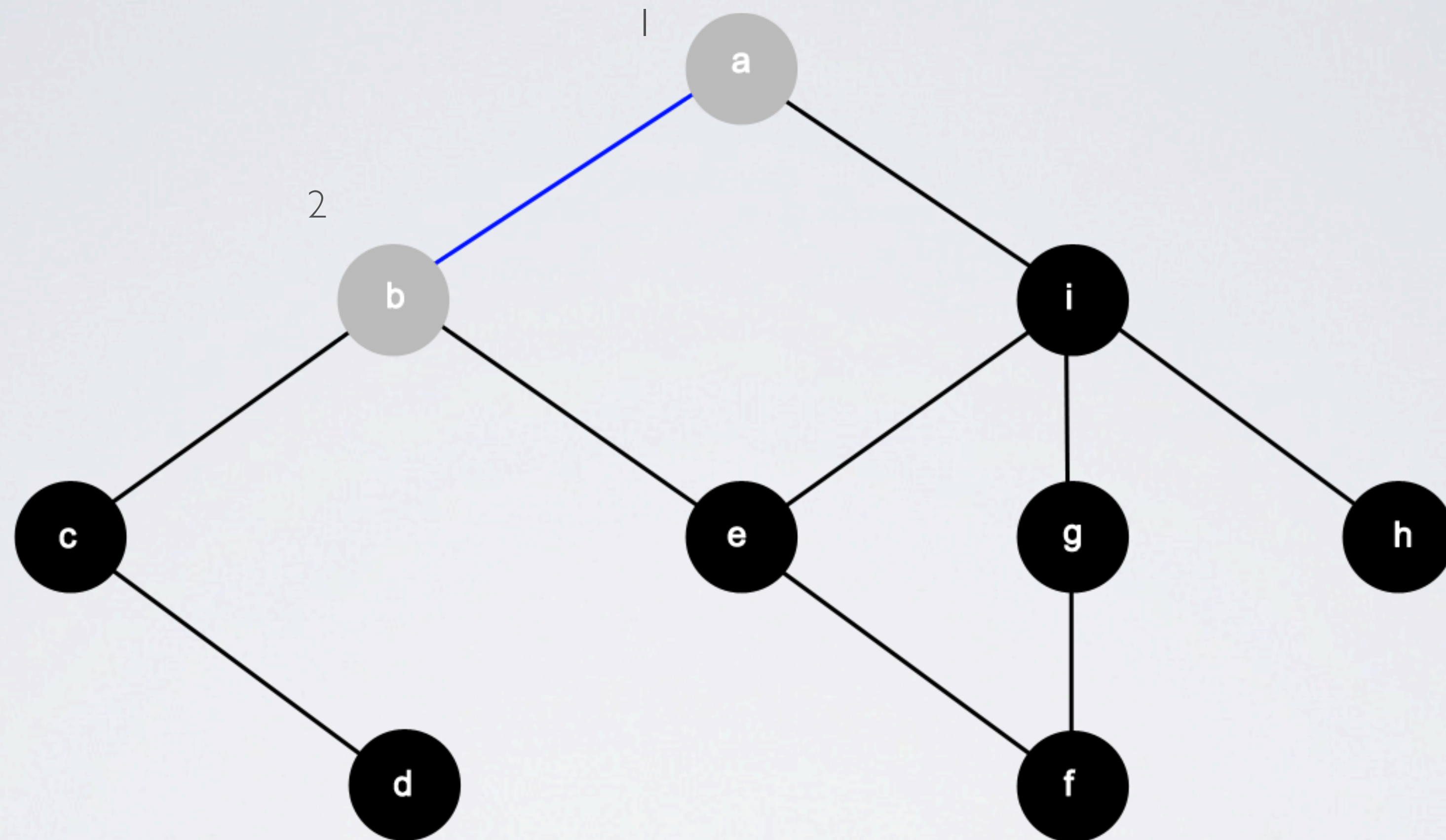
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



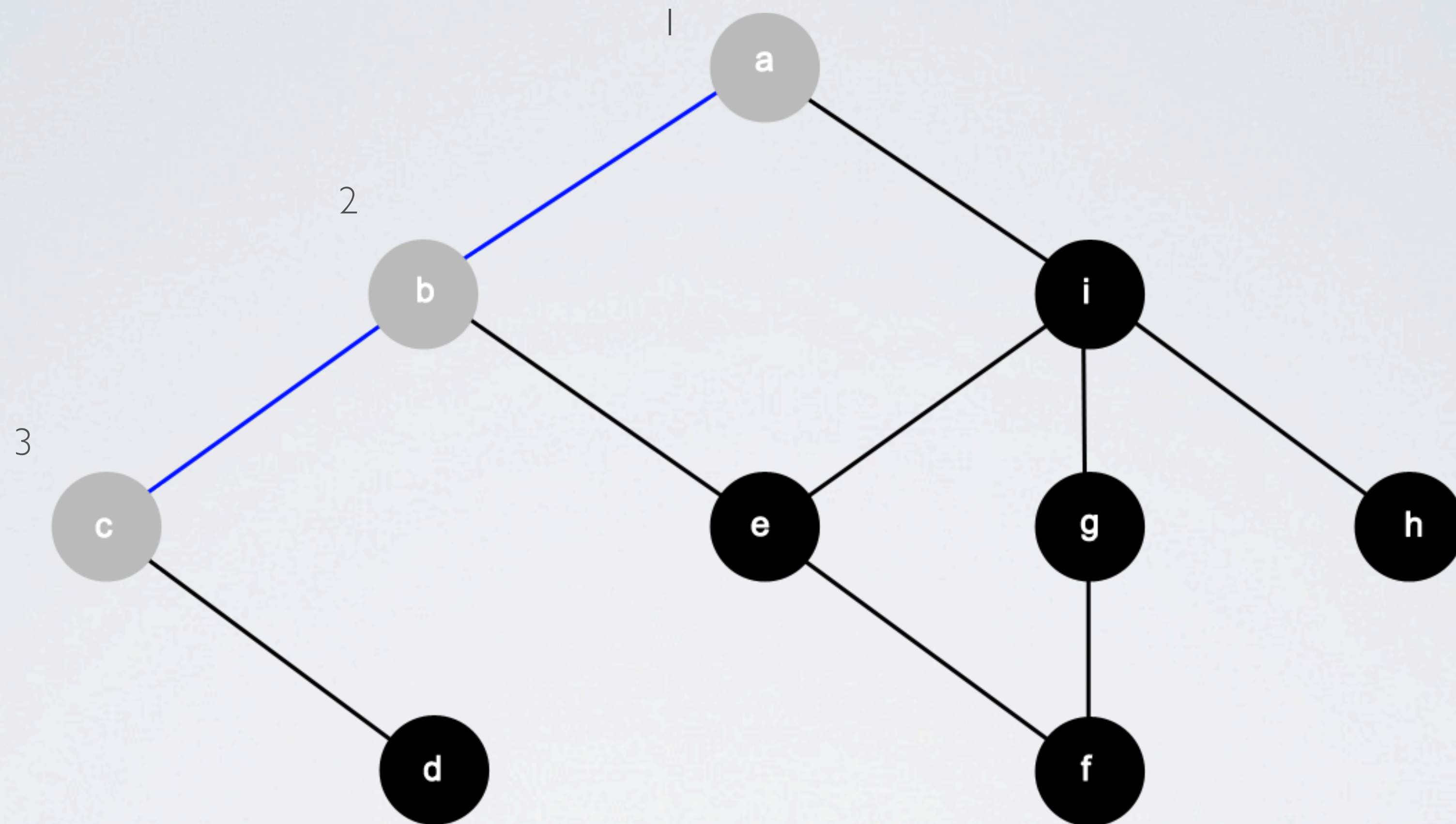
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



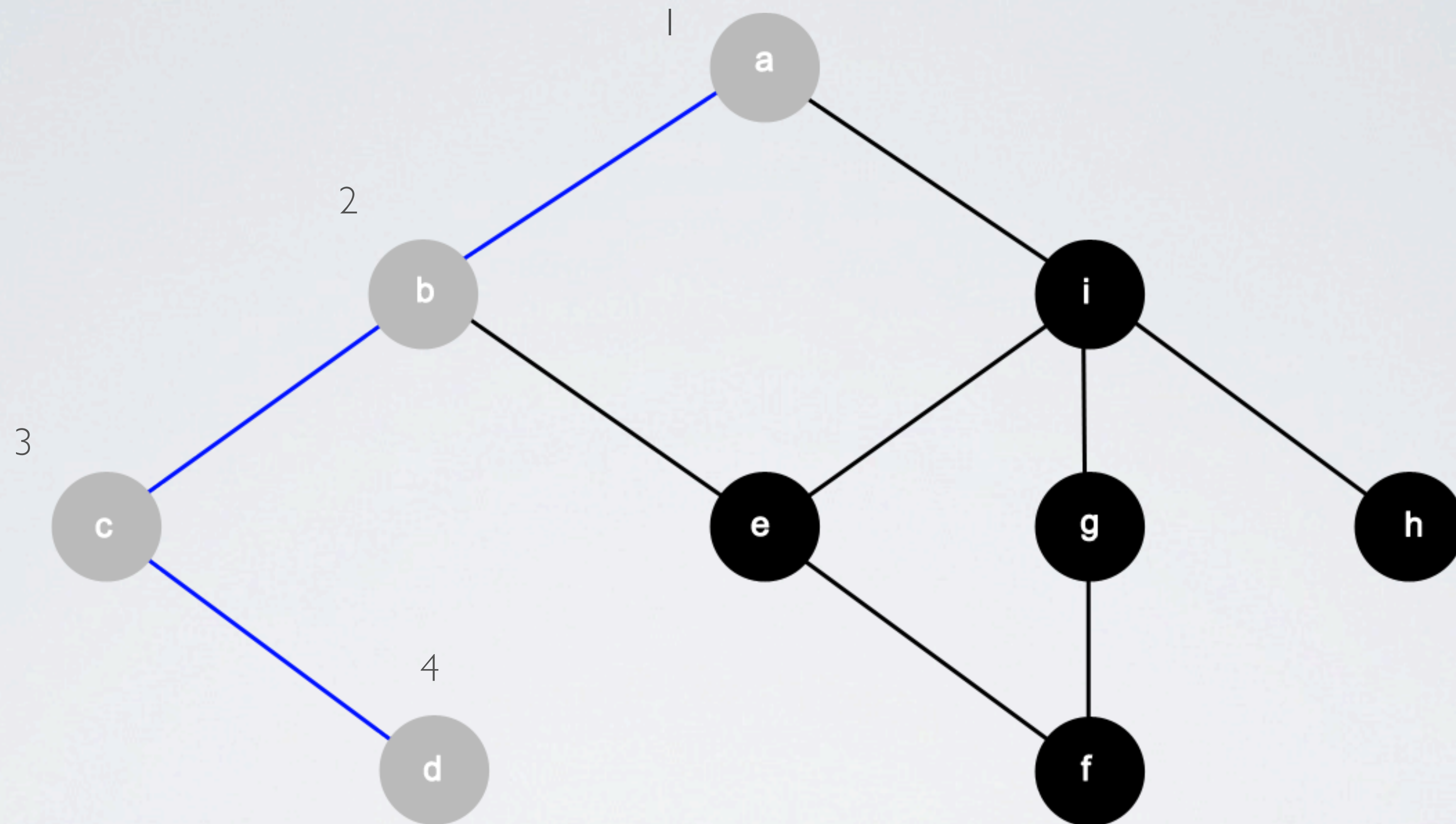
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



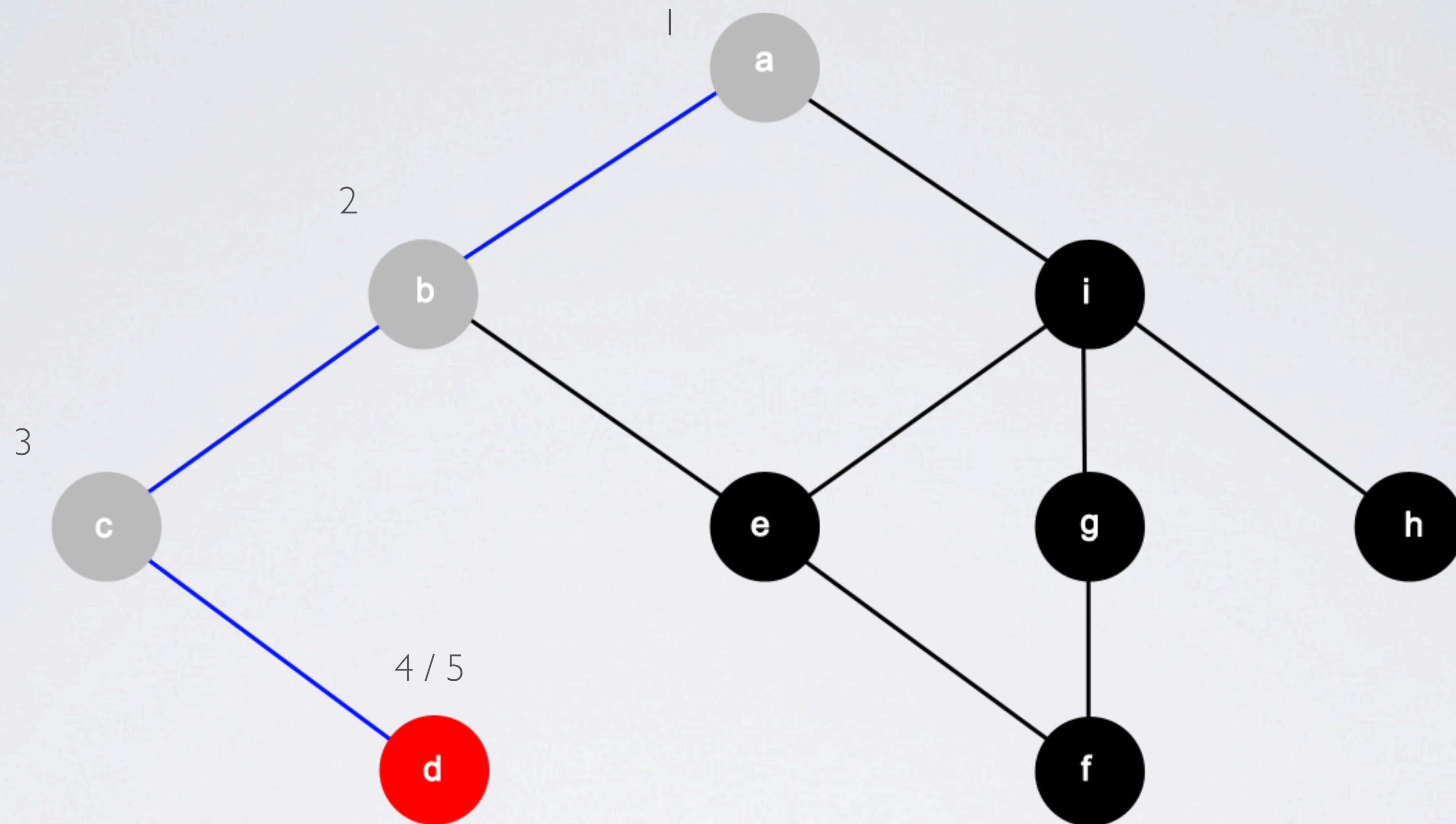
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



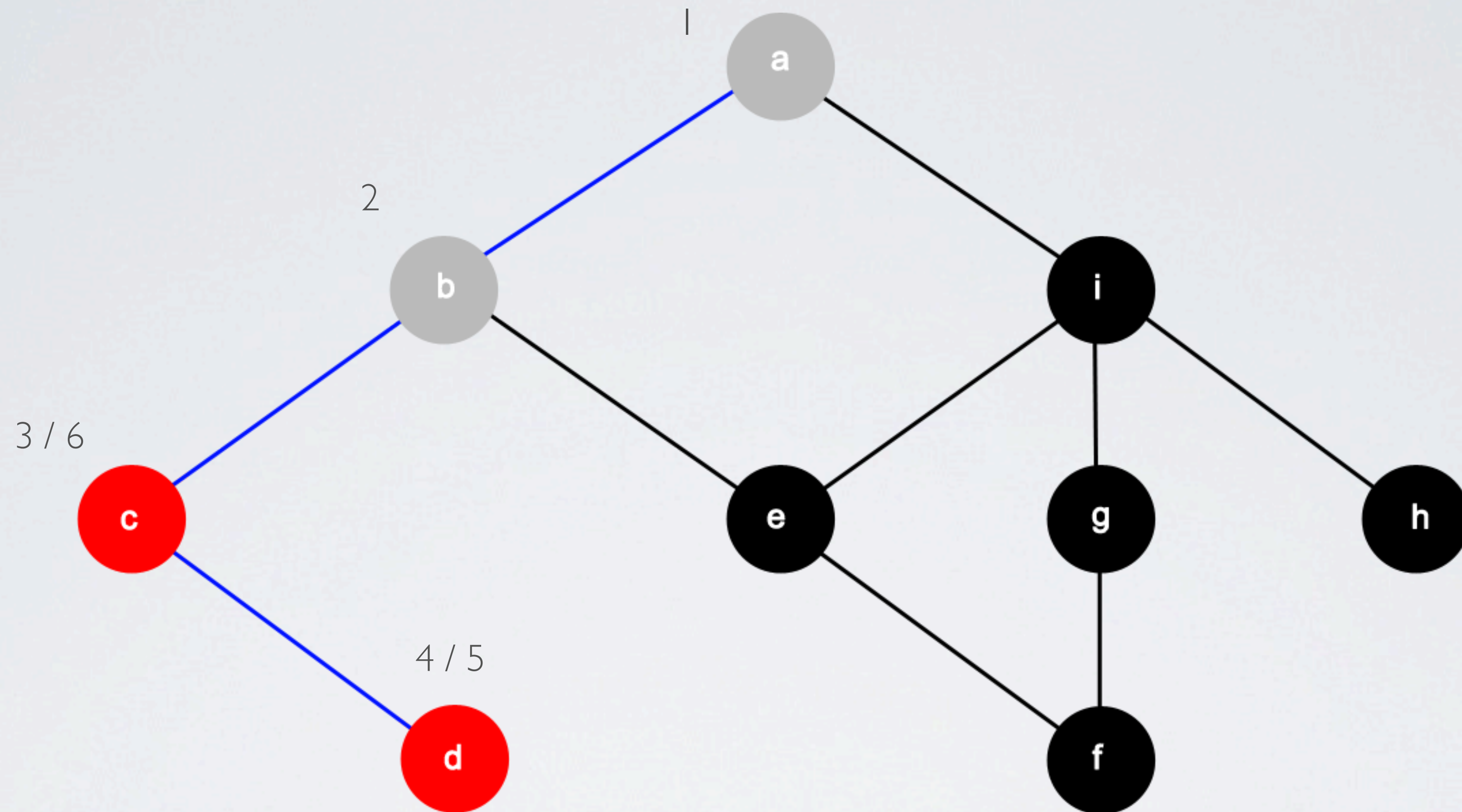
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



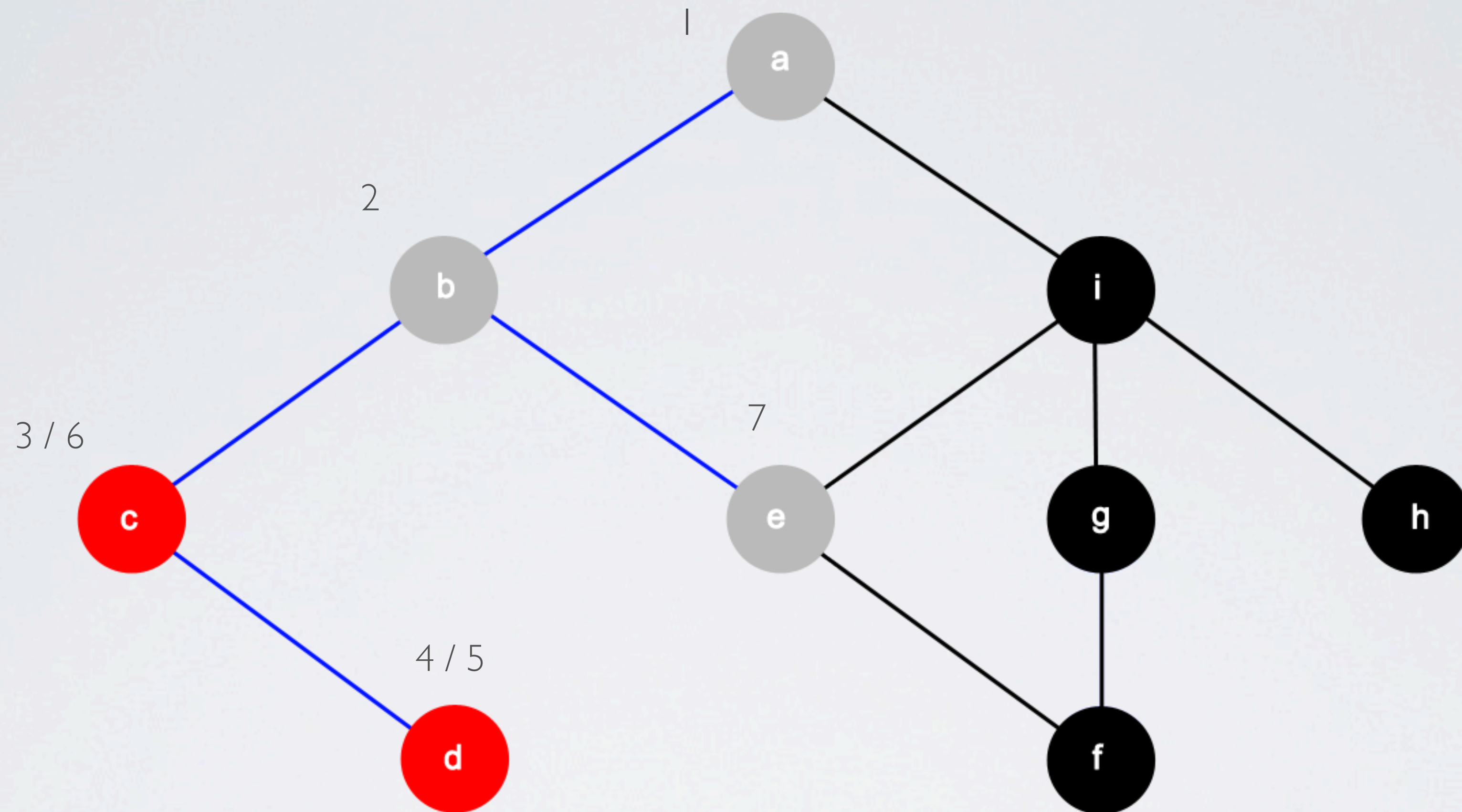
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



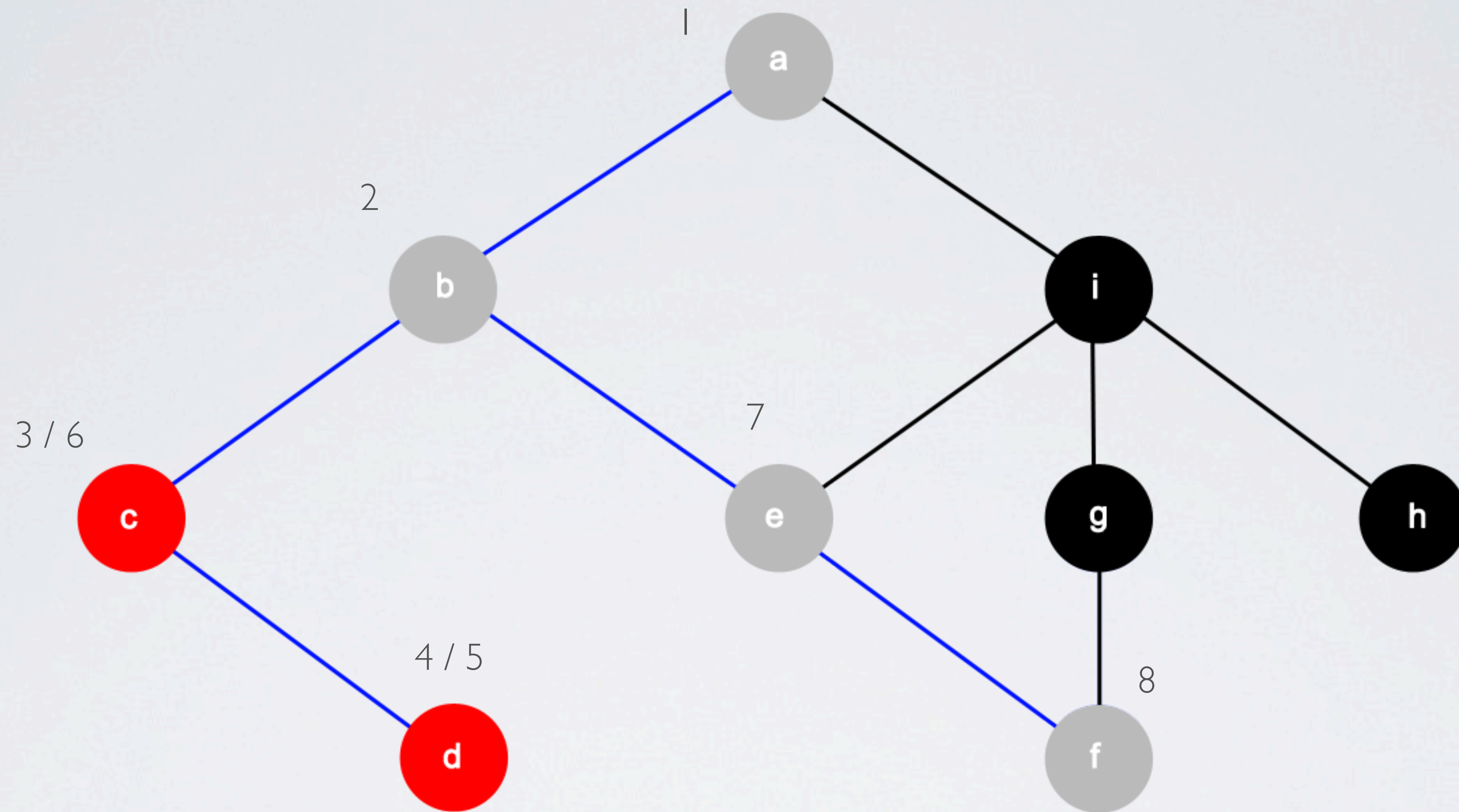
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



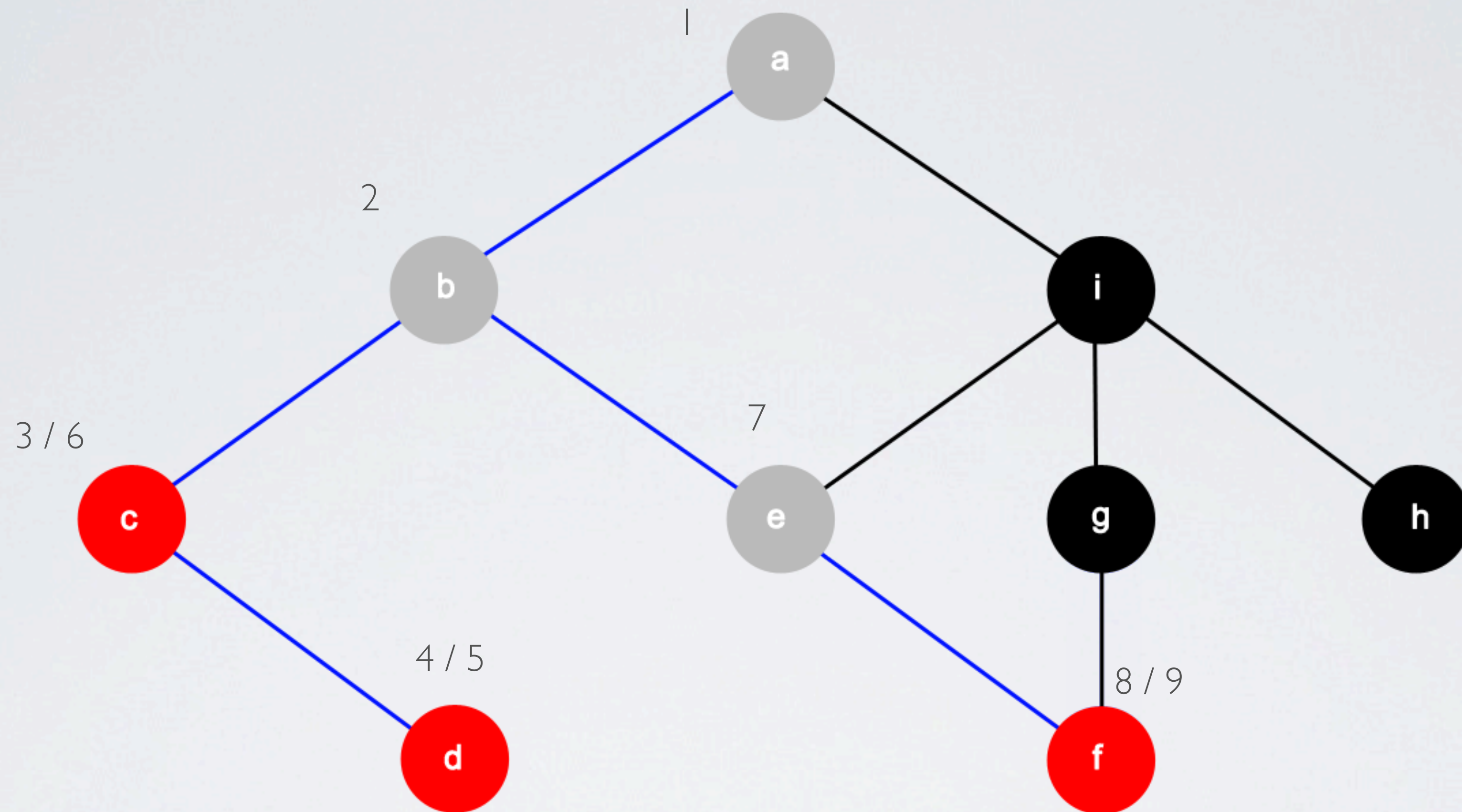
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



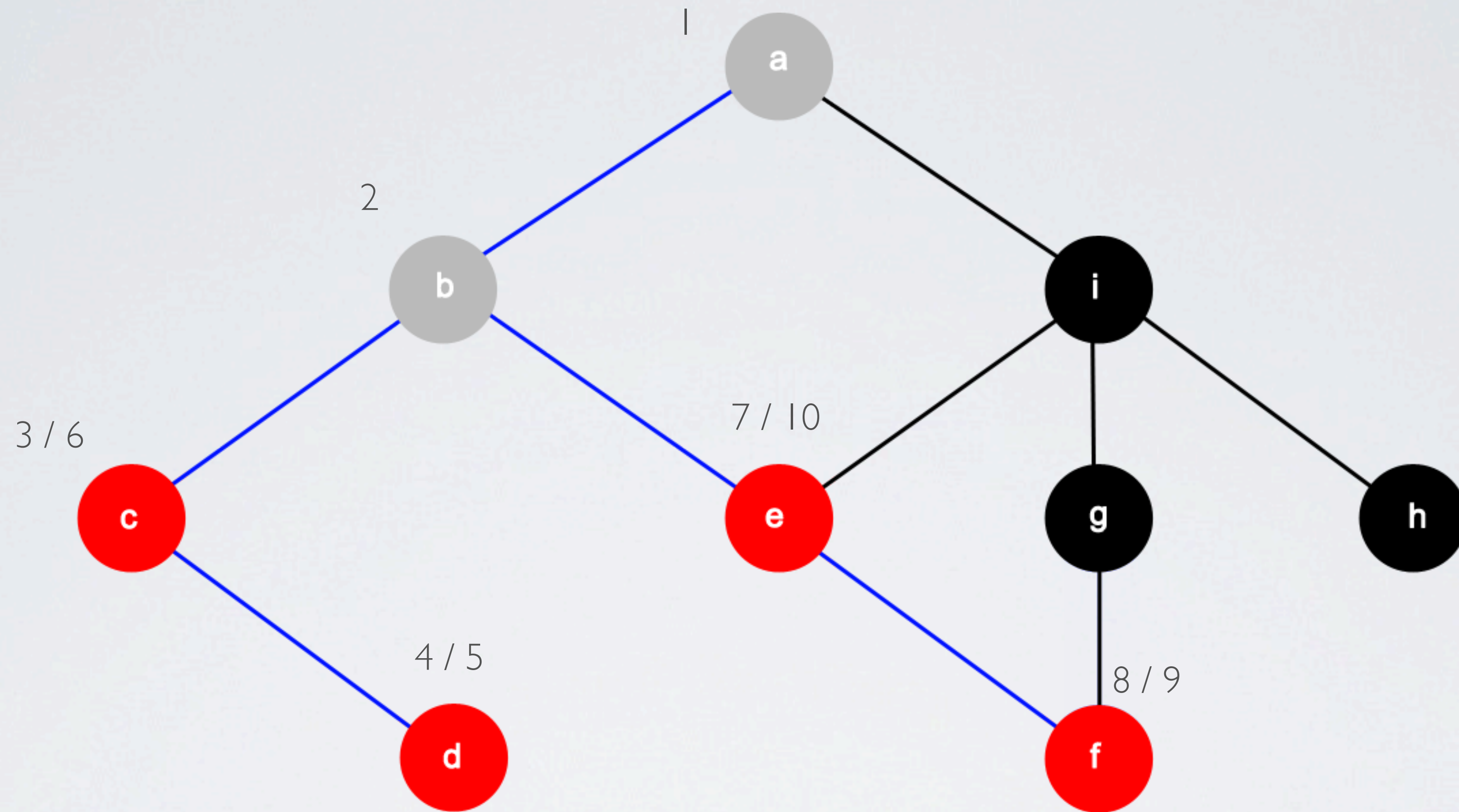
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



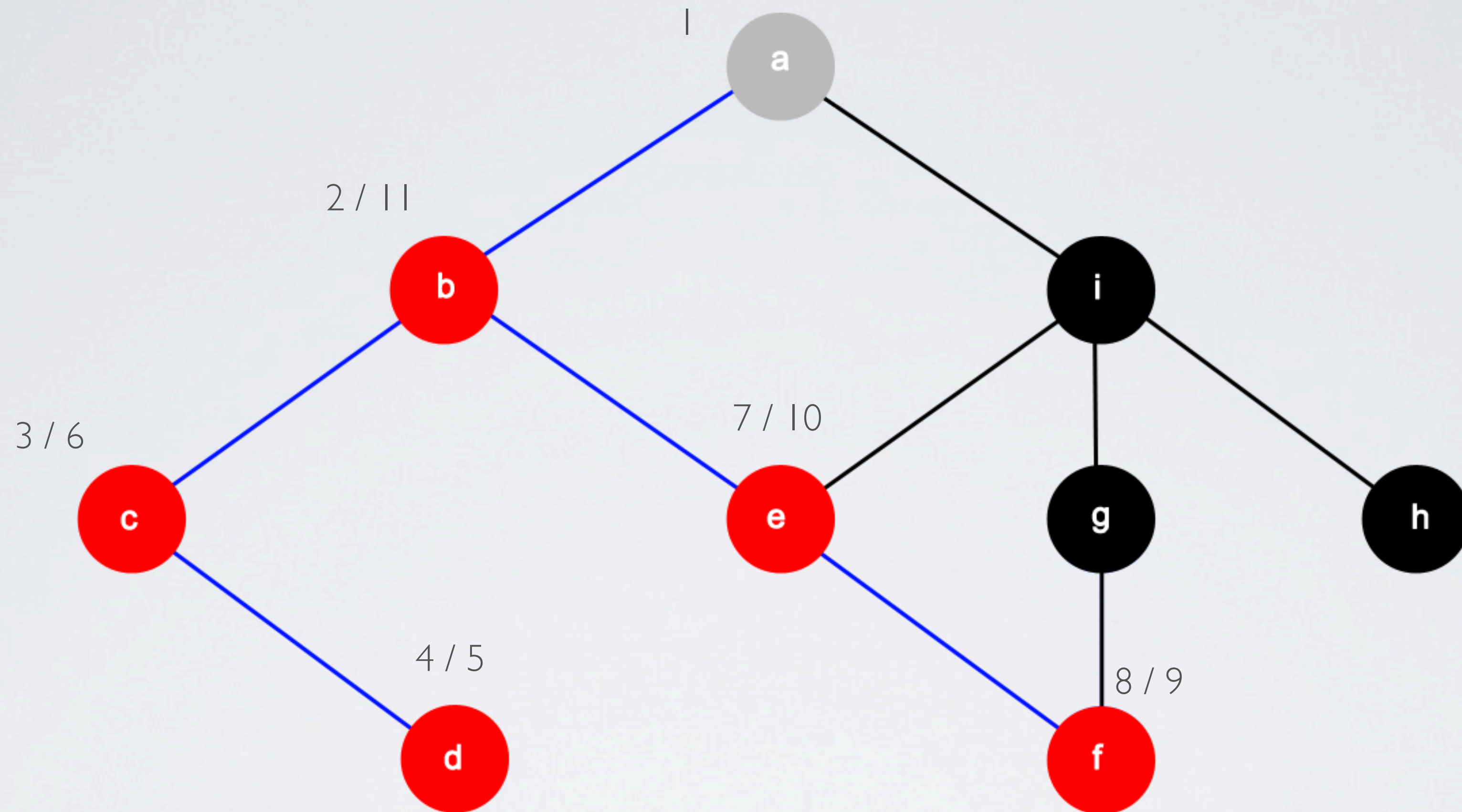
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



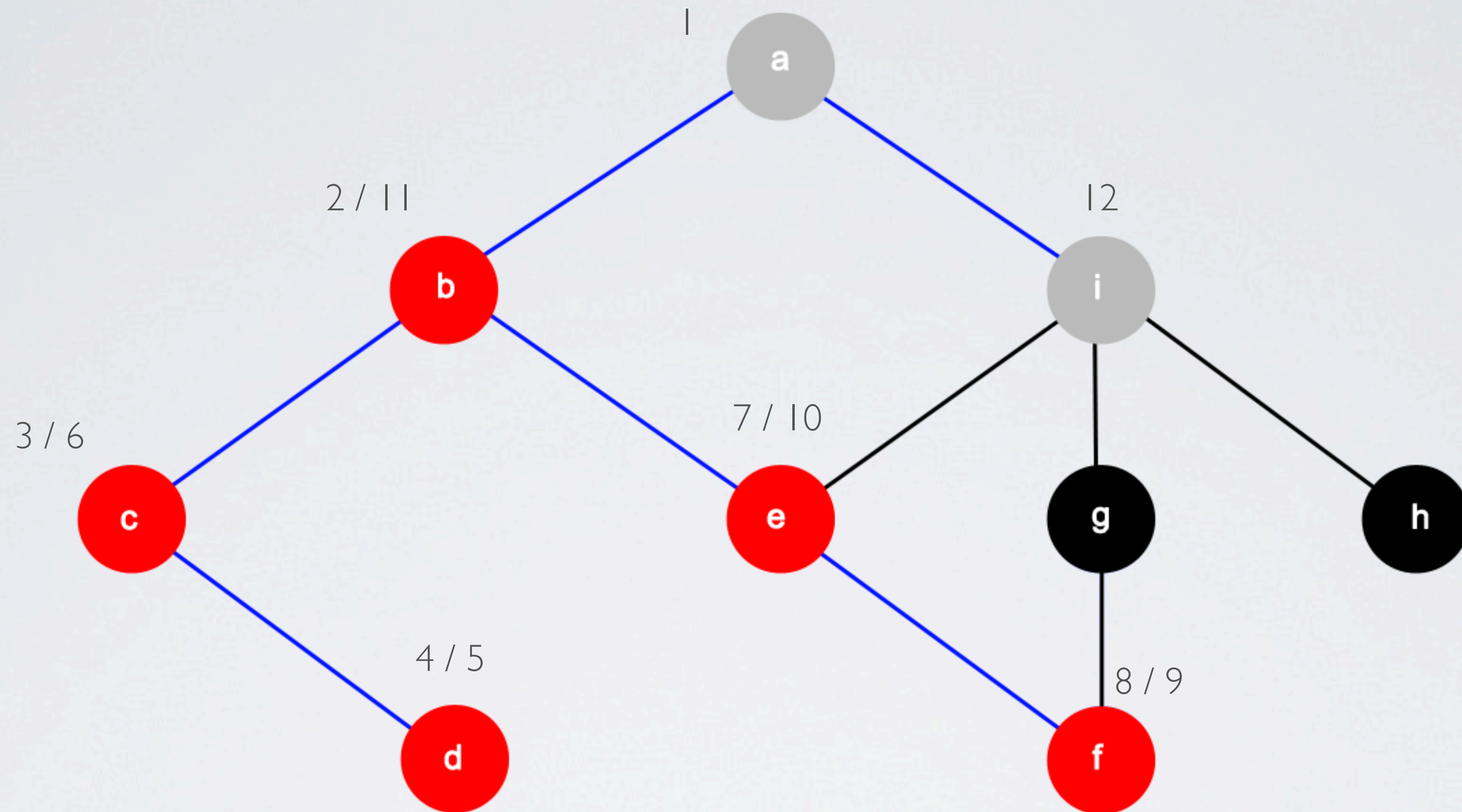
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



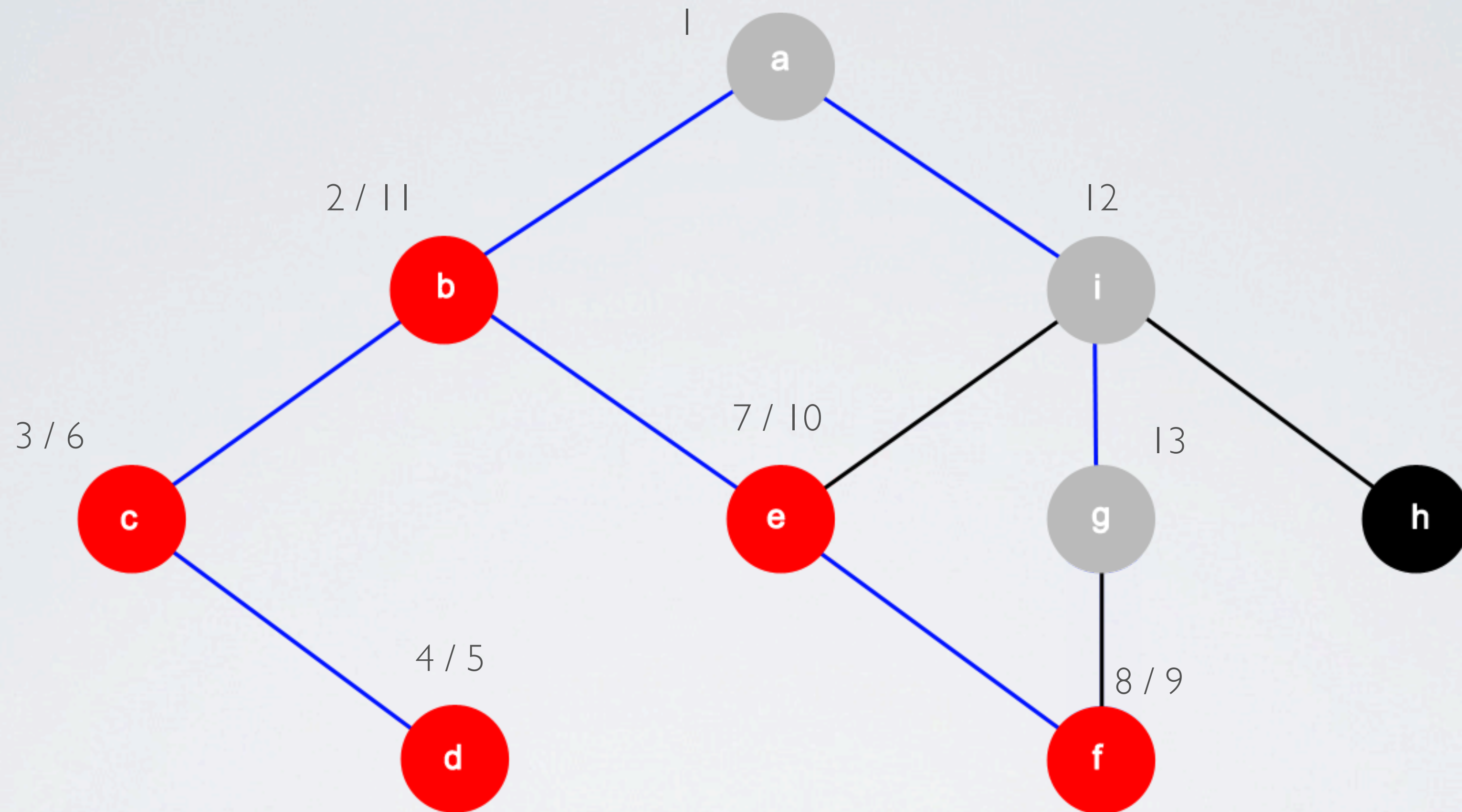
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



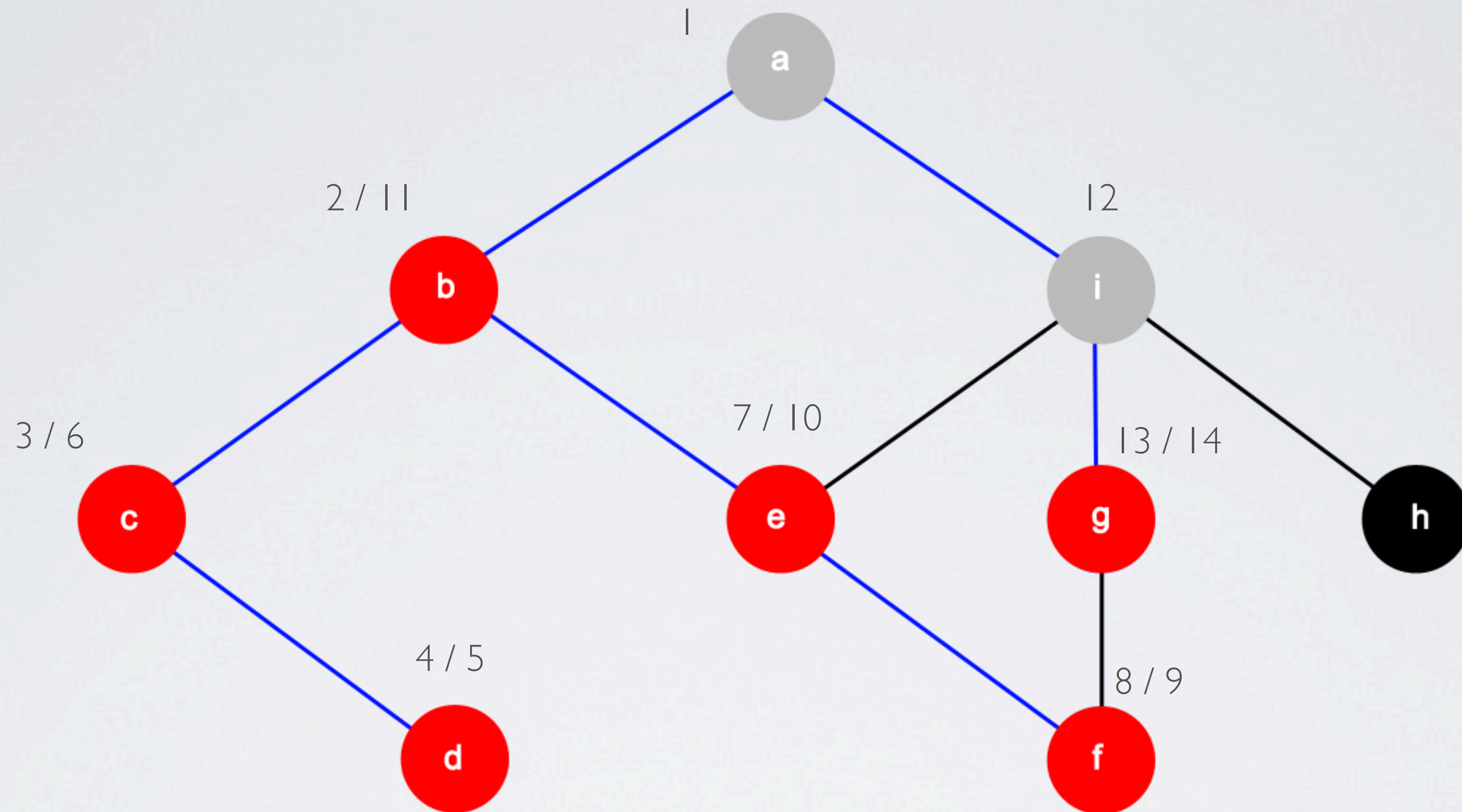
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



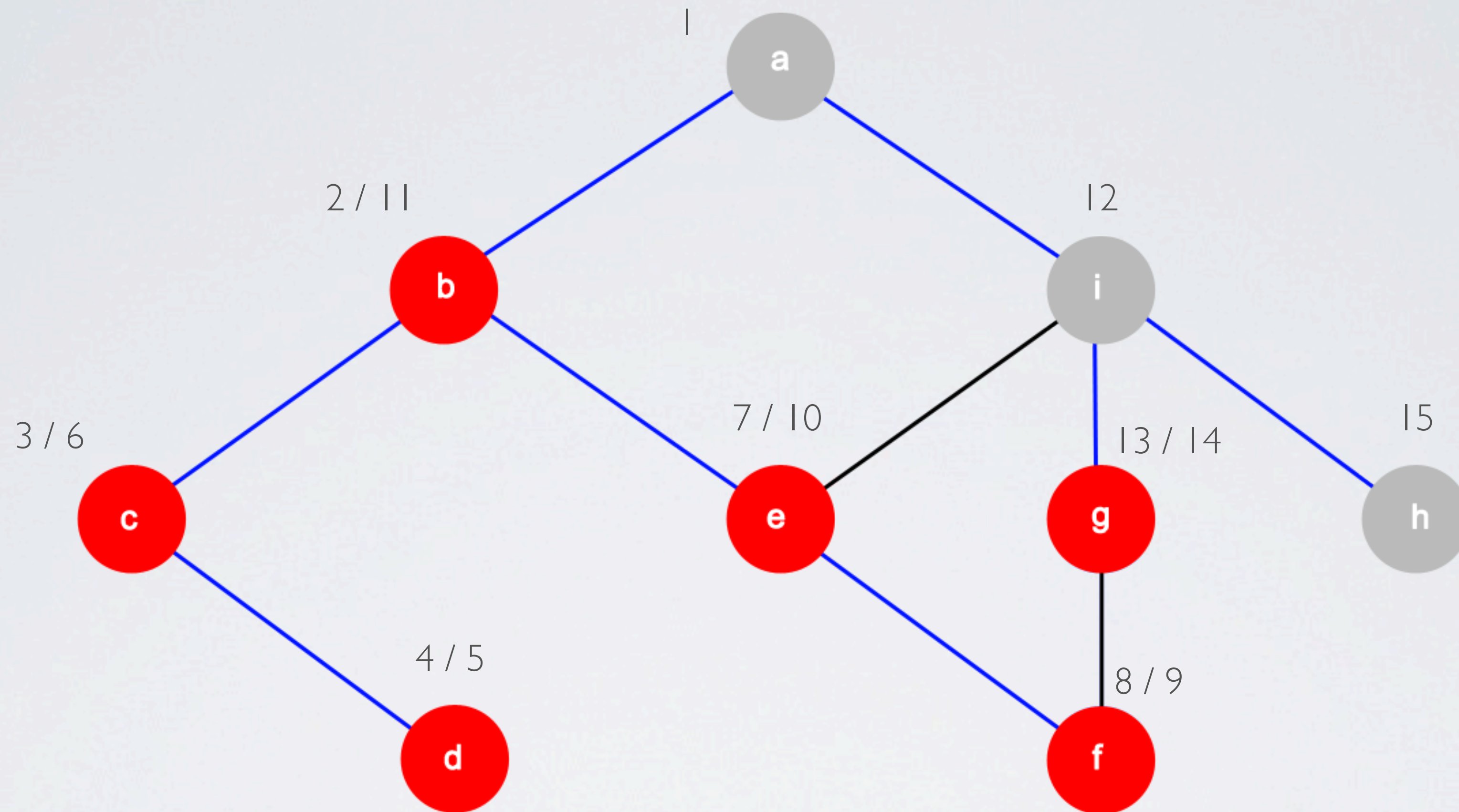
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



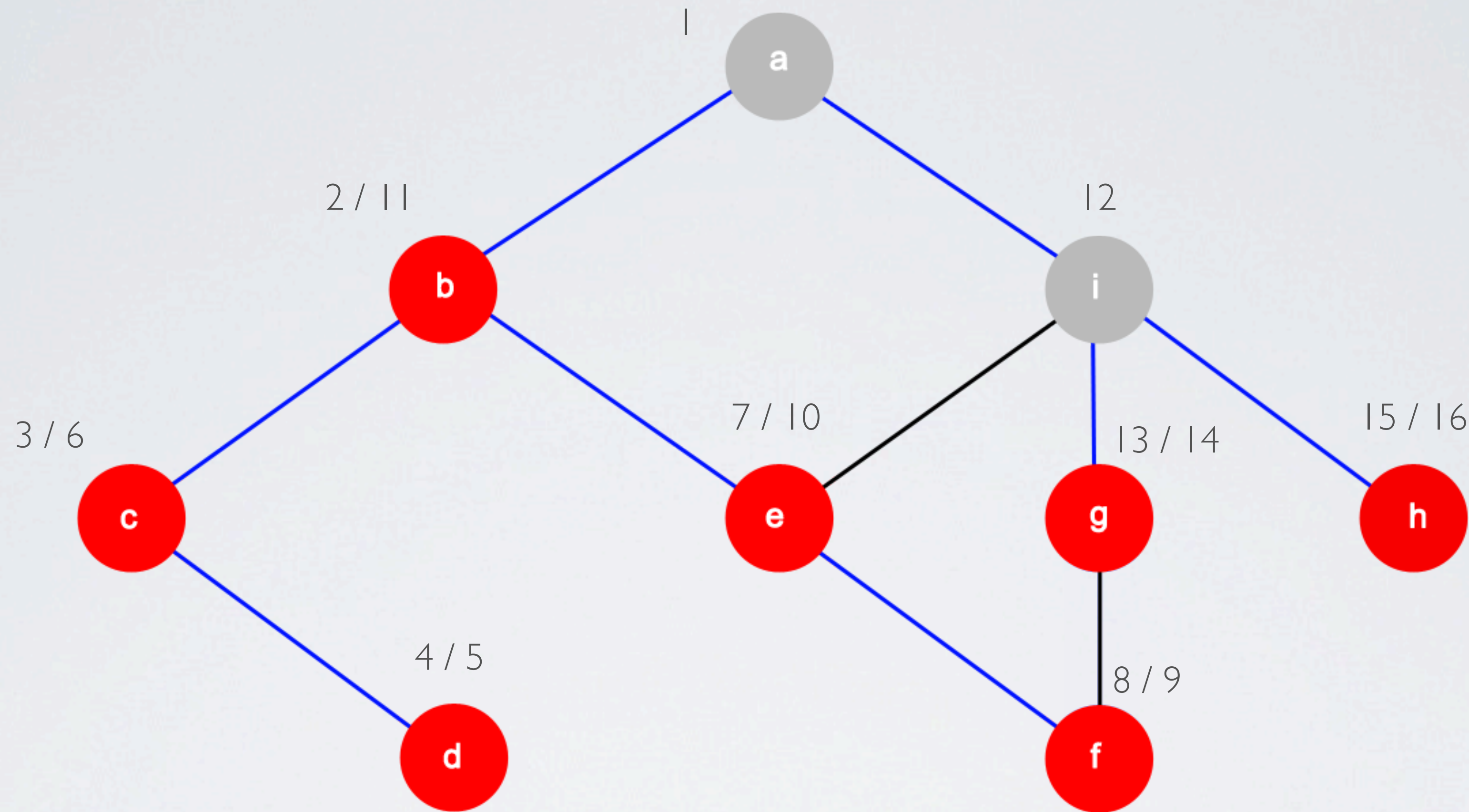
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



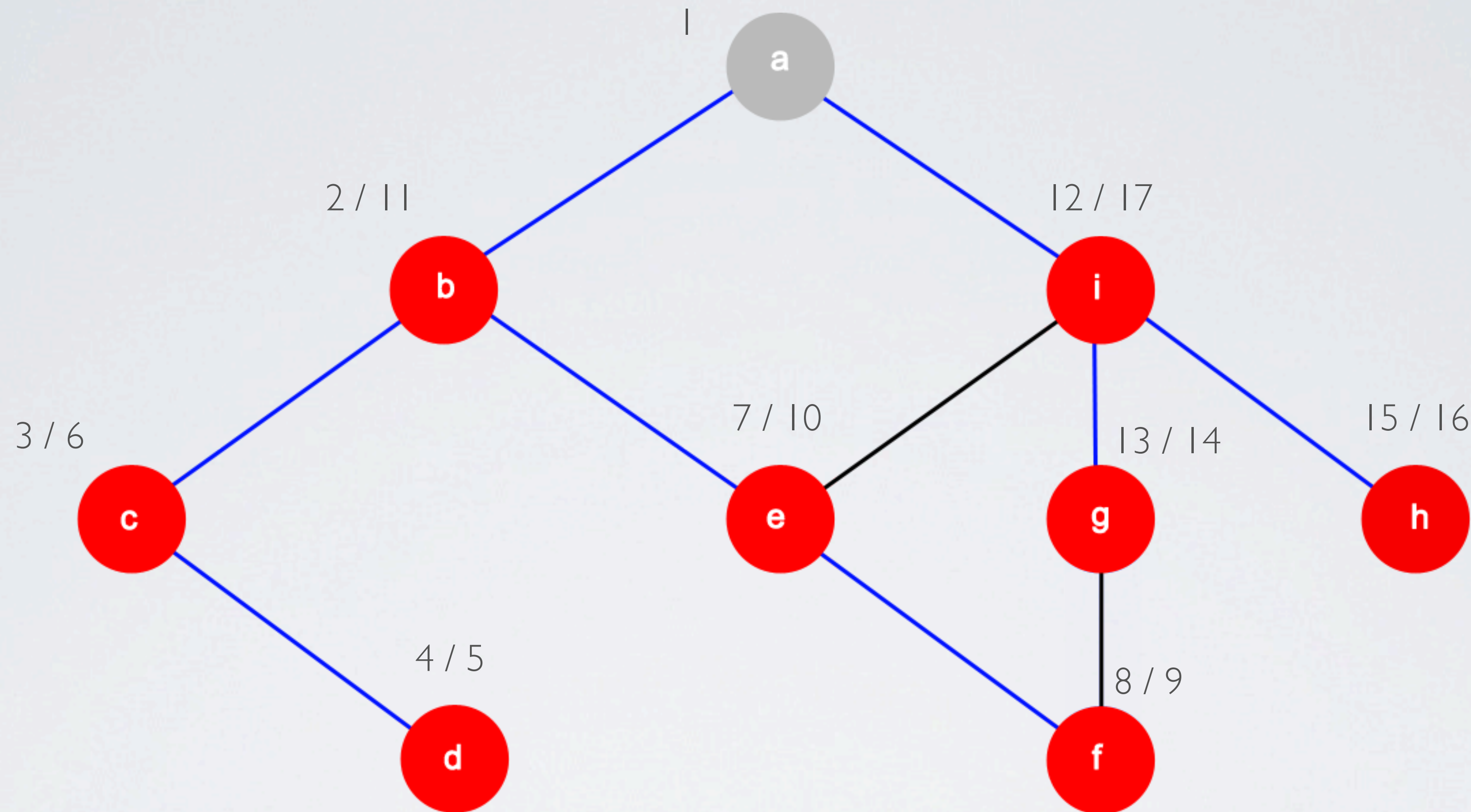
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



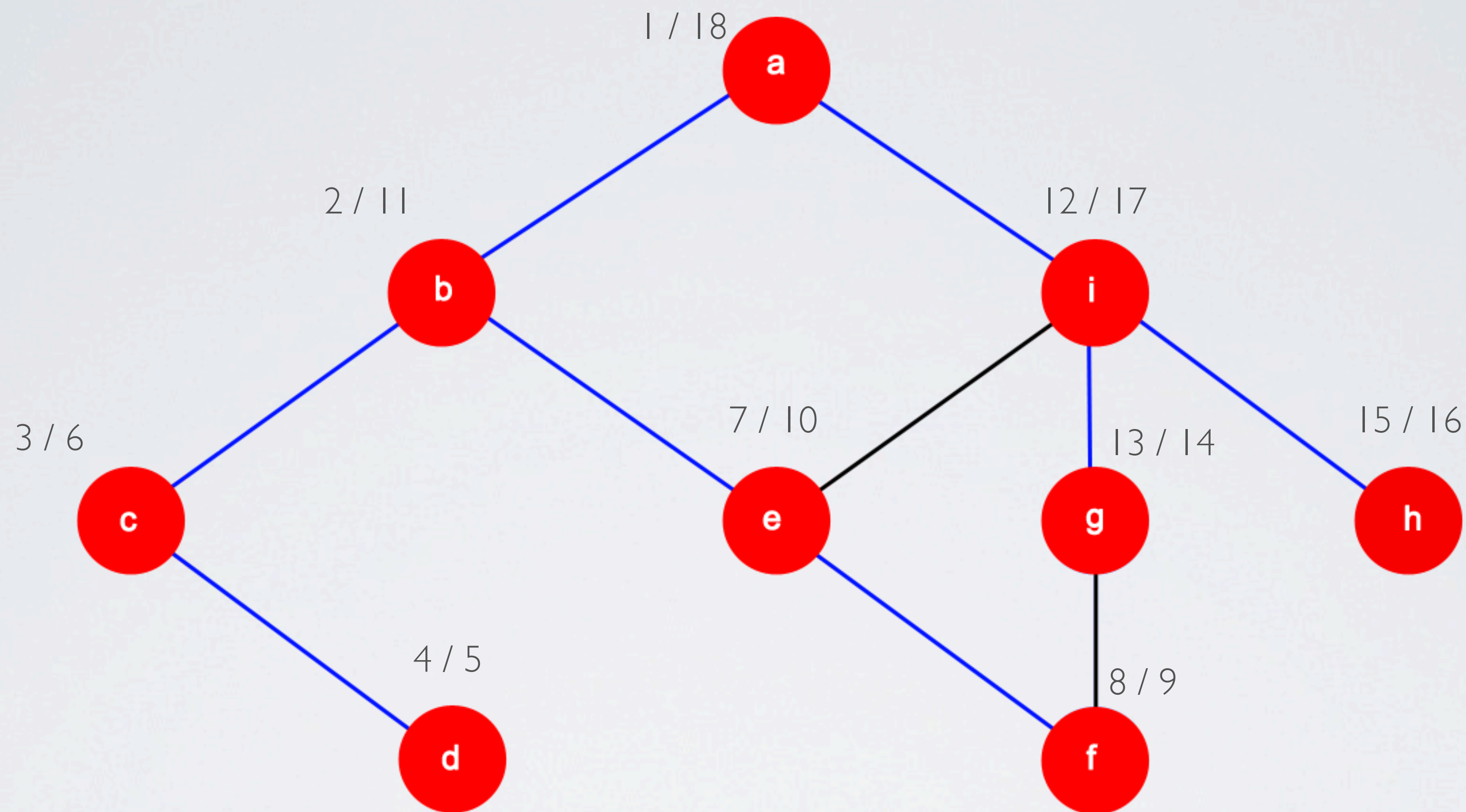
ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS



ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

DFS

ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

```
dfs(graph)
{
  for each node in graph
  {
    color[node] = black
    parent[node] = null
    t1[node] = 0
  }
  time = 0
  for each node in graph
  {
    if(color[node] == black)
      DFSVisit(node)
  }
}
```

```
DFSVisit(node)
{
  color[node] = gray
  time++;    t1[node] = time
  for each v adjacent to node
  {
    if(color[v] == black)
    {
      parent[v] = node
      DFSVisit(v)
    }
  }
  color[node] = red
  time++;    t2[node] = time
}
```


ΔΙΑΣΧΙΣΗ ΚΑΤΑ ΒΑΘΟΣ

Υπάρχουν τρία είδη ακμών:

- Ακμές δέντρου: Οι ακμές που χρησιμοποιήσαμε στον DFS (μπλε χρώμα).
- Πίσω ακμές: Οι ακμές του γράφου οι οποίες συνδέουν ένα παιδί με ένα πρόγονο του στο δέντρο του DFS.
- Μπροστά ακμές: Οι ακμές του γράφου οι οποίες συνδέουν ένα πρόγονο με κάποιο παιδί του στο δέντρο του DFS.
- Ακμές διασταύρωσης: οι υπολοιπες ακμές.

ΕΥΡΕΣΗ ΚΥΚΛΟΥ

Θεώρημα:

Ένα γράφημα είναι ακυκλικό αν και μόνο αν δεν έχει πίσω ακμές.

Συνεπώς για να βρούμε αν ένα γράφημα έχει κύκλο, τρέχουμε τον αλγόριθμο DFS και έπειτα ελέγχουμε τις ακμές που δεν χρησιμοποιήθηκαν αν είναι πίσω ακμές.

ΤΟΠΟΛΟΓΙΚΗ ΔΙΑΤΑΞΗ

ΣΥΝΔΕΔΕΜΕΝΕΣ ΣΥΝΙΣΤΩΣΕΣ

ΙΣΧΥΡΑ ΣΥΝΔΕΔΕΜΕΝΕΣ ΣΥΝΙΣΤΩΣΕΣ

ΒΙΒΛΙΟΓΡΑΦΙΑ

- www.wikipedia.org
- Σημειώσεις κ. Φωτάκη από το μάθημα Αλγόριθμοι και Πολυπλοκότητα ΣΗΜΜΥ (<http://www.corelab.ntua.gr/courses/algorithms/>)
- Σημειώσεις κ. Saad Mneimneh (<http://www.cs.hunter.cuny.edu/~saad/courses/493.55/notes/note5.pdf>)