BFS

```
BFS(G, s)
for each vertex u ∈ V                          O(|V|)
        color(u) = white
        π(u) = Nil
        d(u) = ∞
color(s) = gray
π(s) = Nil
d(s) = 0
Q as an empty queue
Enqueue(Q, s)
while (Q ≠ ϕ)
        u = Dequeue(Q)
        for each vertex v ∈ Adj(u)             all vs 2|E| times
                if color(v) = white
                        color(v) = gray
                        π(v) = u
                        d(v) = d(u) + 1
                        Enqueue(Q, v)
        color(u) = black
```

overall time:

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

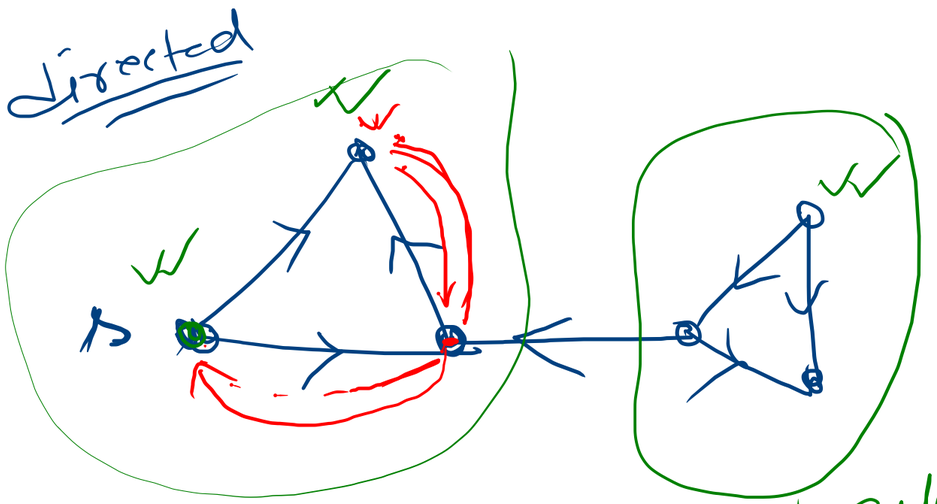
$= O(|V| + |E|)$

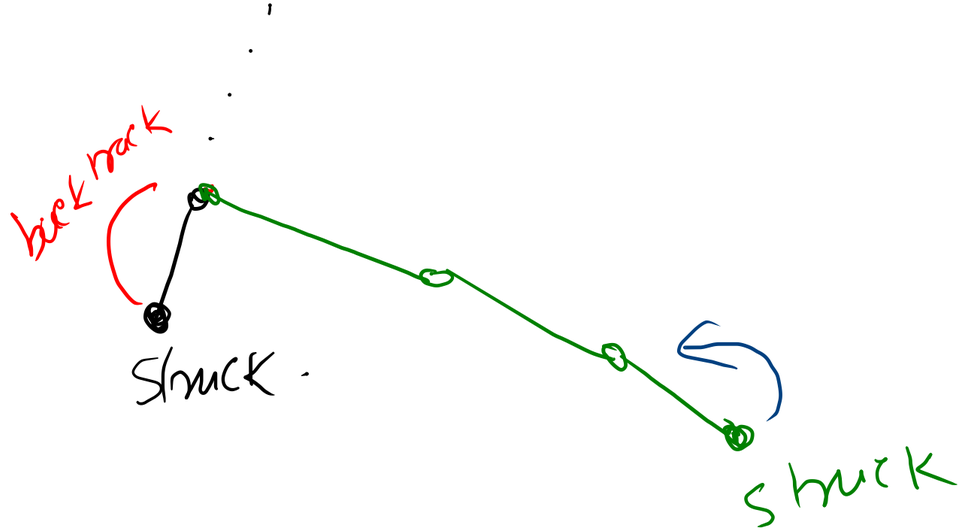
$(\pi(v), v)$

Depth - first Search (DFS)

s - discover time

f - finish time



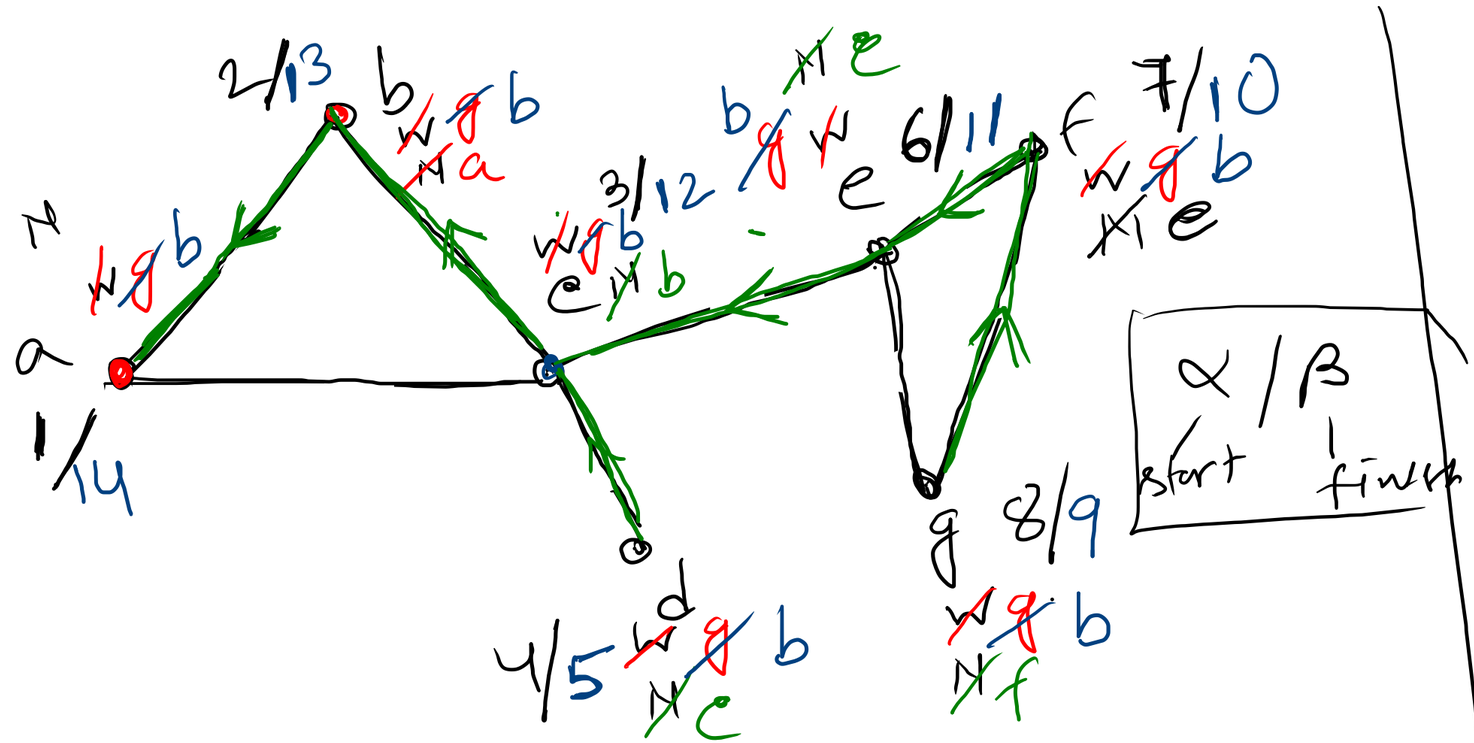


it gives

forest : collection of trees.

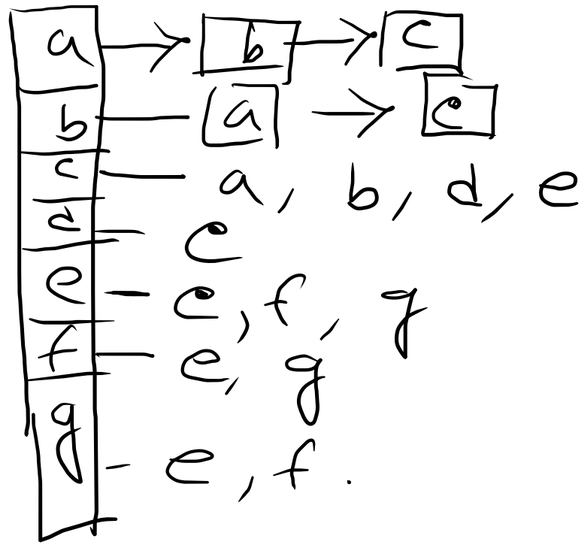
```
DFS(G)
for each vertex u ∈ V
        color(u) = white
        π(u) = Nil
time = 0
for each vertex u ∈ V
        if color(u) = white
                DFS-visit(G, u)
```


```
DFS-visit(G, u)
time = time + 1
s(u) = time
color(u) = gray
for each vertex w ∈ Adj(u)
        if color(w) = white
                π(w) = u
                DFS-visit(G, w)
time = time + 1
f(u) = time
color(u) = black.
```

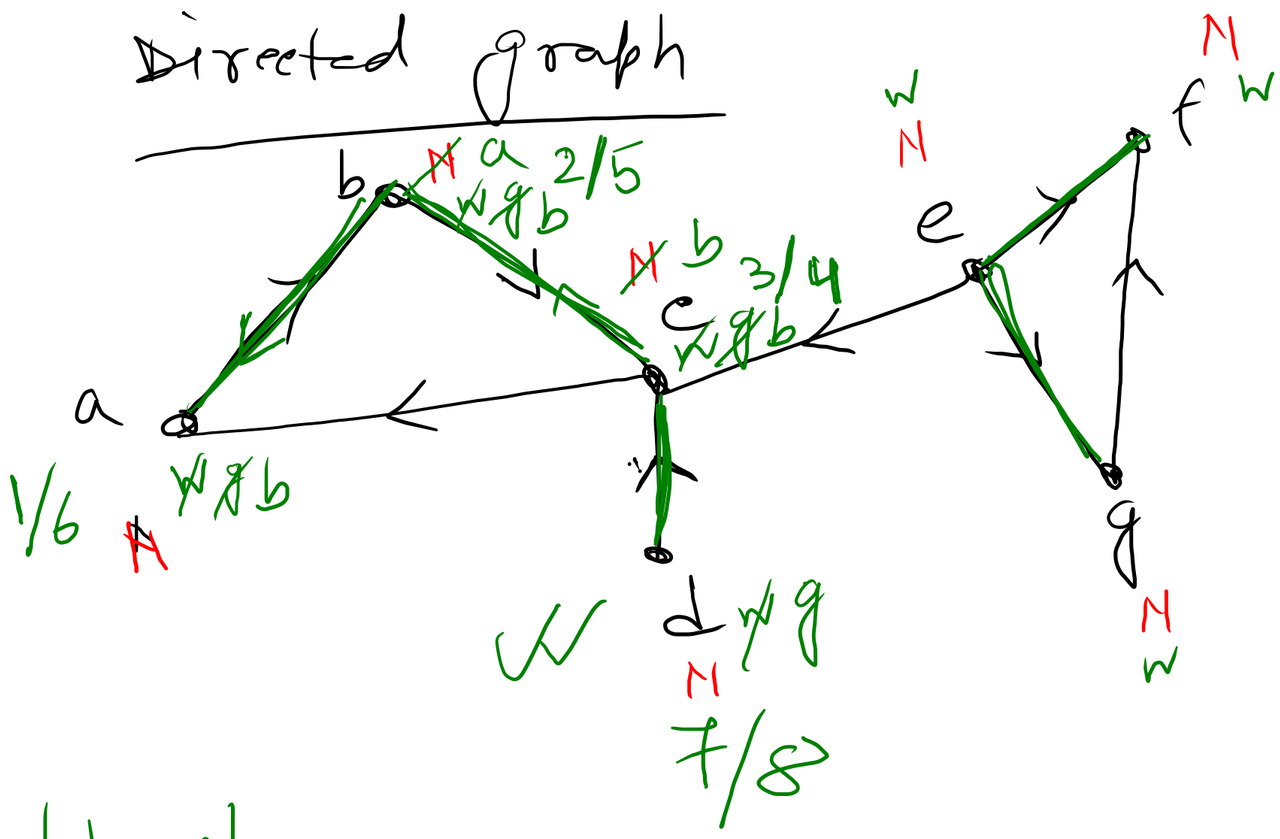


d/f
start finish

Adjacency list representation



Directed graph



Adjacency list

a - b

b - c

c - a

d - c

e - c, f, g

f -

g - f