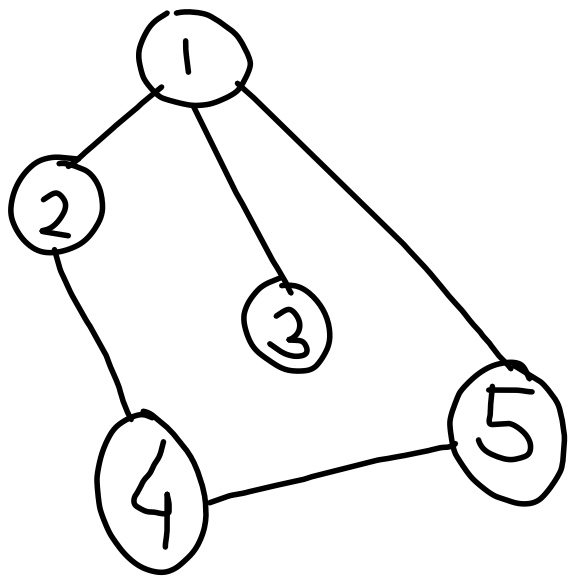
Graph Algorithms

Graph Representation :



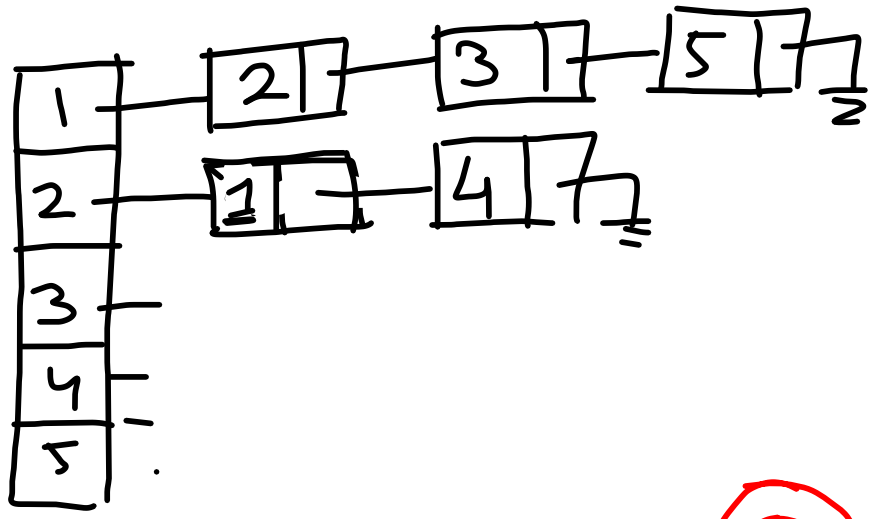
$G = \{(1,2), (1,3), (1,5), (2,4), (4,5)\}$

Space → $O(|E|)$

Q Is there an edge between 3 and 5?

$O(|E|)$

A



Space = $O(n+|E|)$

Adjacency List

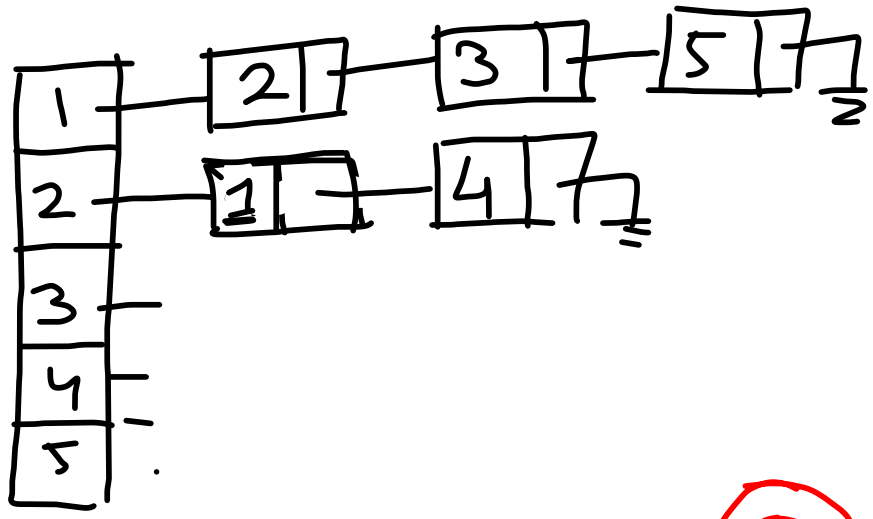
Q

$O(\Delta)$ → max degree

$\approx O(n)$

$|V| = n$

A



Space = $O(n+|E|)$

Adjacency List

Q

$O(\Delta)$ → max degree

$\approx O(n)$

$|V| = n$

Adjacency Matrix

A

	1	2	3	4	5
1	0	1	1	0	1
2	-			-	
3			-		
4		-			
5					

Space → $O(n^2)$

Q → $O(1)$

A[2,3] = ?

1 → edge

0 → no edge

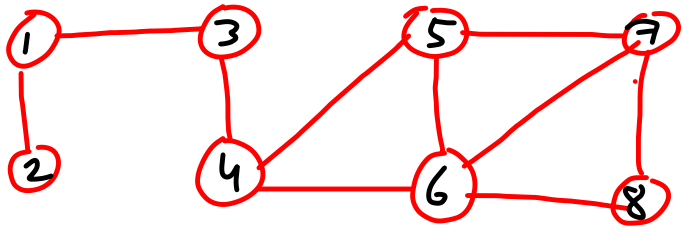
- Sparse → Adjacency List
- Dense → Adjacency Matrix

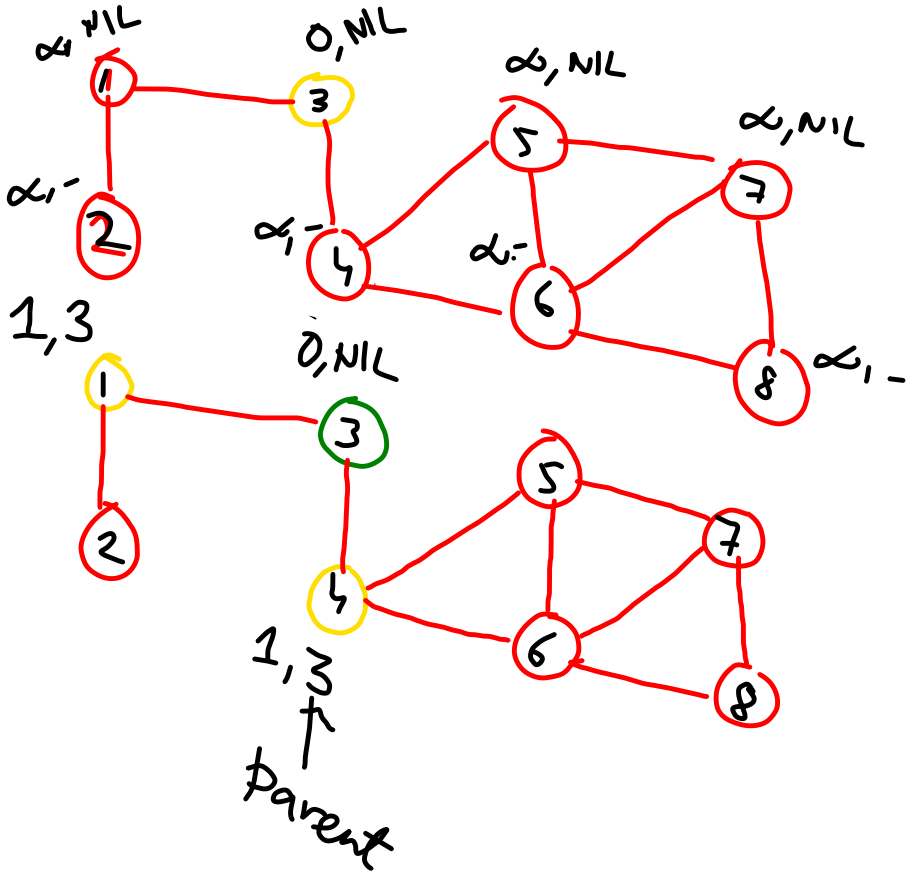
Graph Search Algorithms

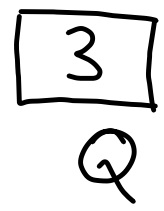
Applications :

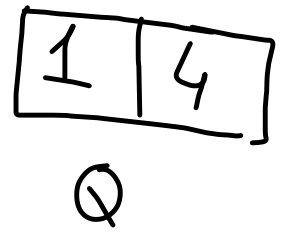
- Web crawling
- GPS navigation
- FB Friend finder
- N/W Broadcast
- Garbage Collection

BFS (Breadth First Search)









Search

- mindistance from s to t
- path from s to t (min dist)
- Cycle detection

BFS(G, s)

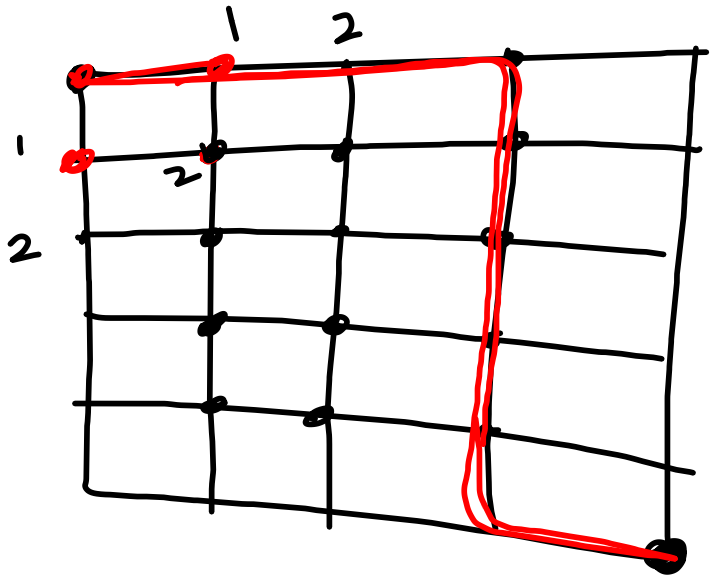
$\forall u \in V - \{s\}$

Init

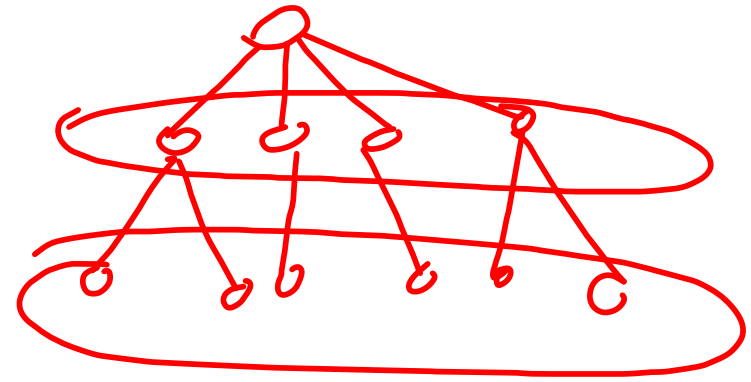
```
u.color = RED
u.dist = ∞
u.parent = NIL
s.color = YELLOW
s.dist = 0
s.parent = NIL
Q = ∅
Enqueue(Q, s)
```

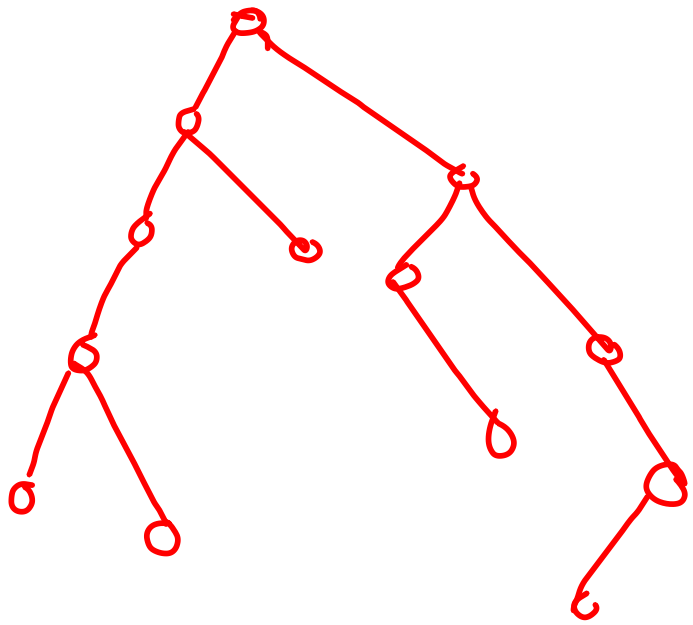
$\Theta(|V|+|E|)$

```
while (Q ≠ ∅)
    u = DEQUEUE(Q)
    ∀ v ∈ Adj[u]
        if (v.color == RED)
        {
            v.color = yellow
            v.dist = u.dist + 1
            v.parent = u
            Enqueue(Q, v)
        }
    u.color = GREEN
```



BFS





DFS

DFS(G)

$\forall u \in V$

$u.color = RED$

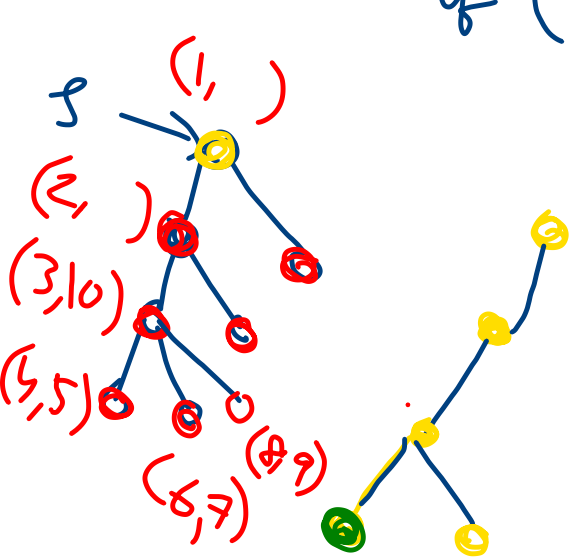
$u.parent = NIL$

$time = 0$

$\forall u \in V$

if $(u.color == RED)$

DFS_VISIT(G,u)



DFS_VISIT(G, u)

$time = time + 1$

$u.start = time$

$u.color = Yellow$

$\forall v \in Adj[u]$

if $(v.color == Red)$

$v.parent = u$

DFS_VISIT(G, v)

$u.color = GREEN$

$time = time + 1$

$u.finish = time$

Topological Sort

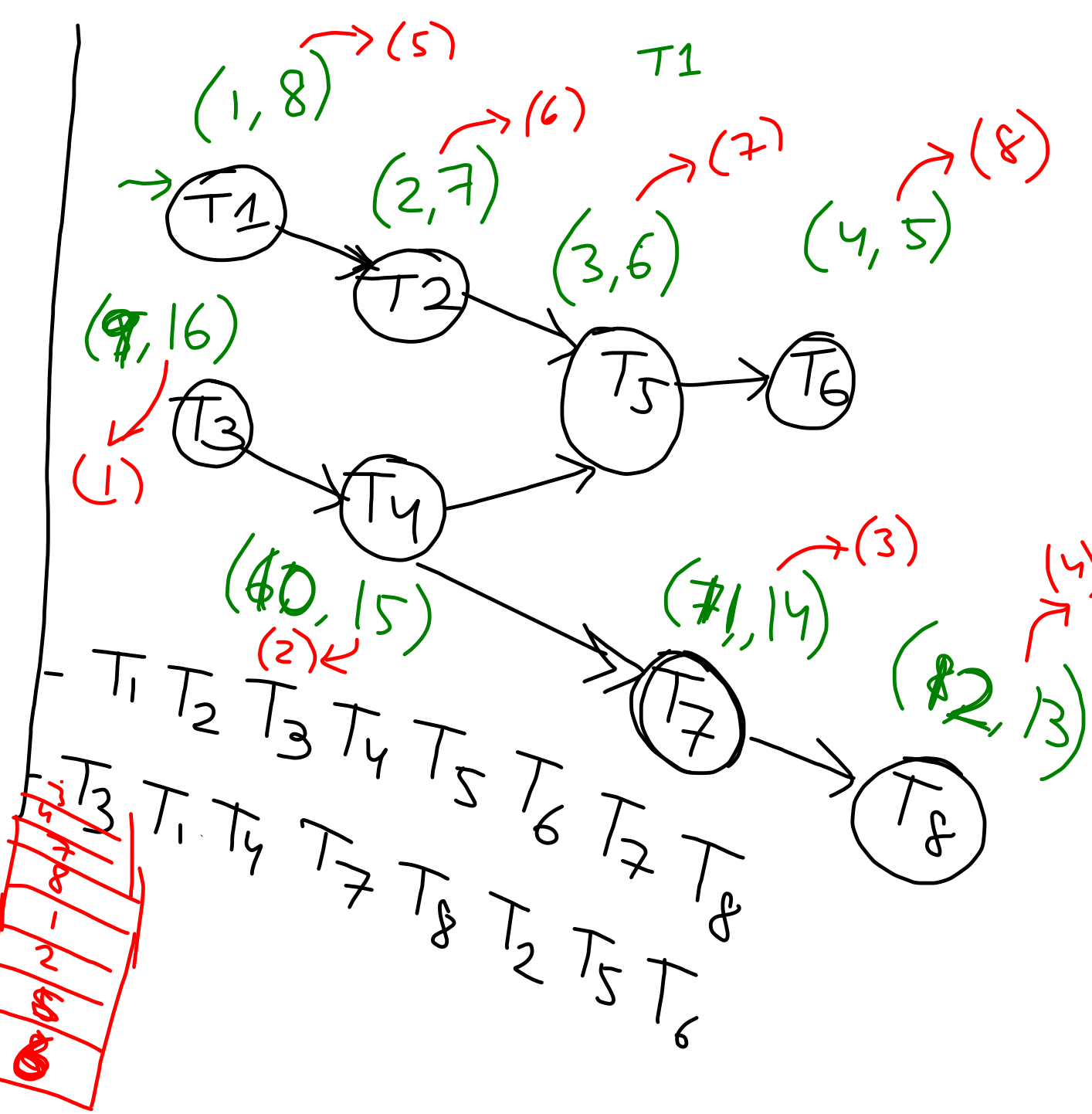
L Linear Ordering

(One such ordering)

DAG

↓

Direct Acyclic Graph



Topological Sort

L Linear Ordering

(One such ordering)

DAG

↓

Direct Acyclic Graph