Trees ✓

Dfs

Bfs

Vertices, edges, neighbours

Adjacency list

Adjacency Matrix

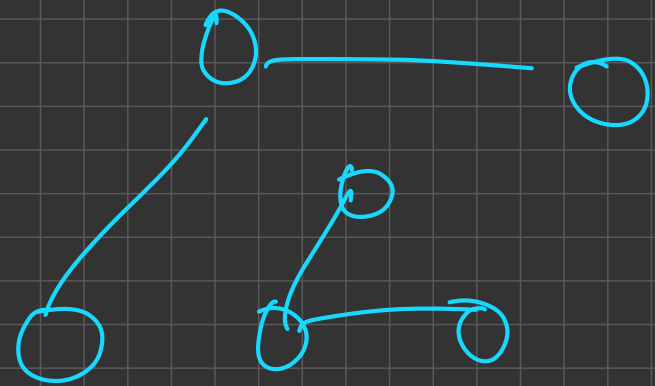
Graphs Class 1

Priyansh Agarwal

Graph

A set of nodes connected via edges

Connected, acyclic graph



Types of Graphs

weighted + Directed Graph

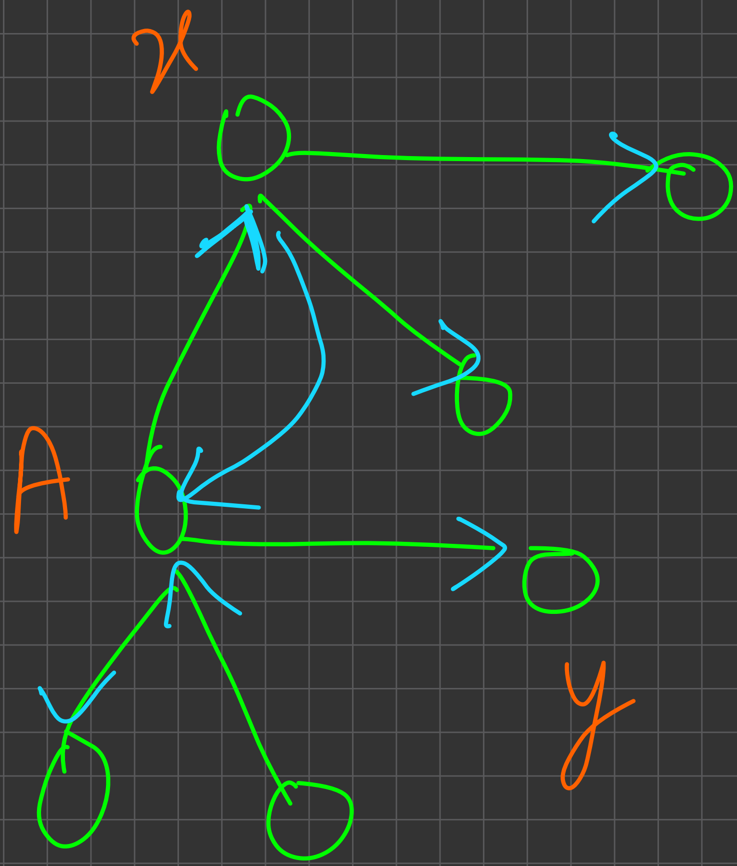
- Undirected vs Directed
- Unweighted vs Weighted
- Cyclic + Acyclic

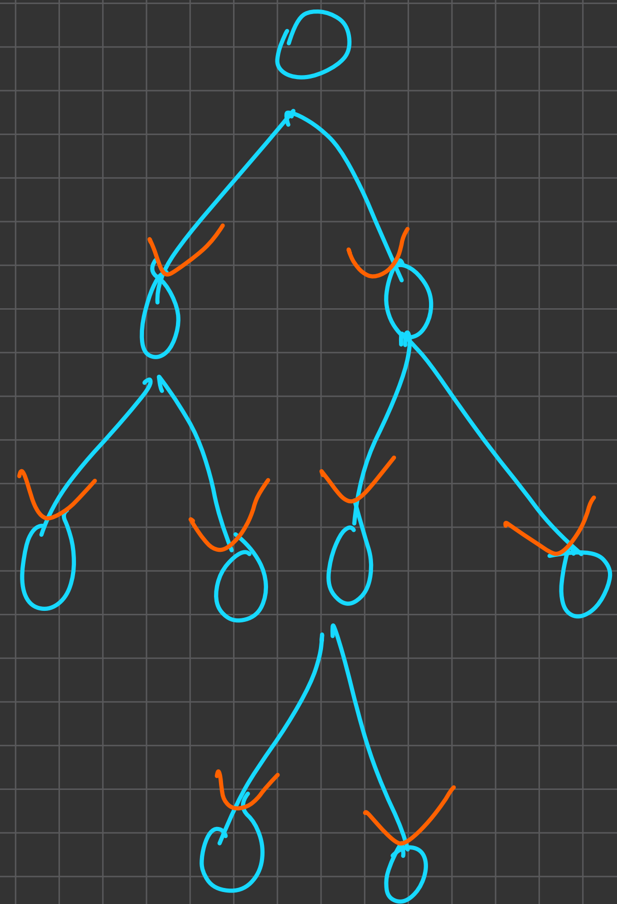
if there exist more than one path b/w any 2 nodes

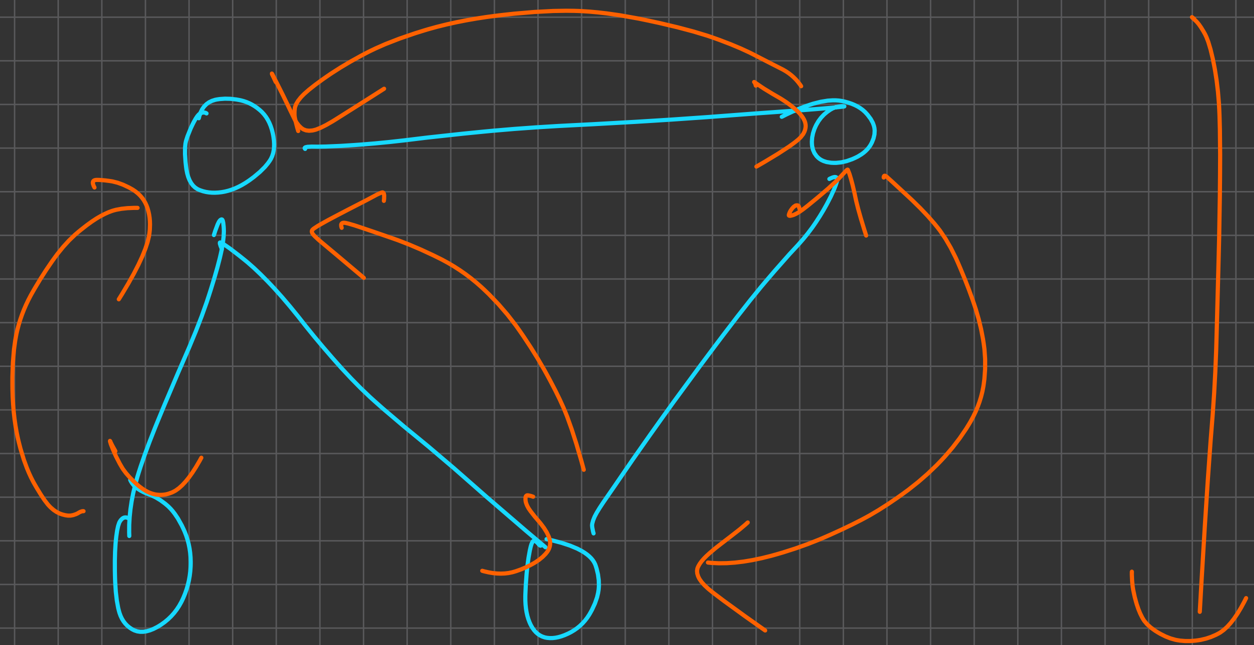
- Connected + Disconnected
- Complete graph

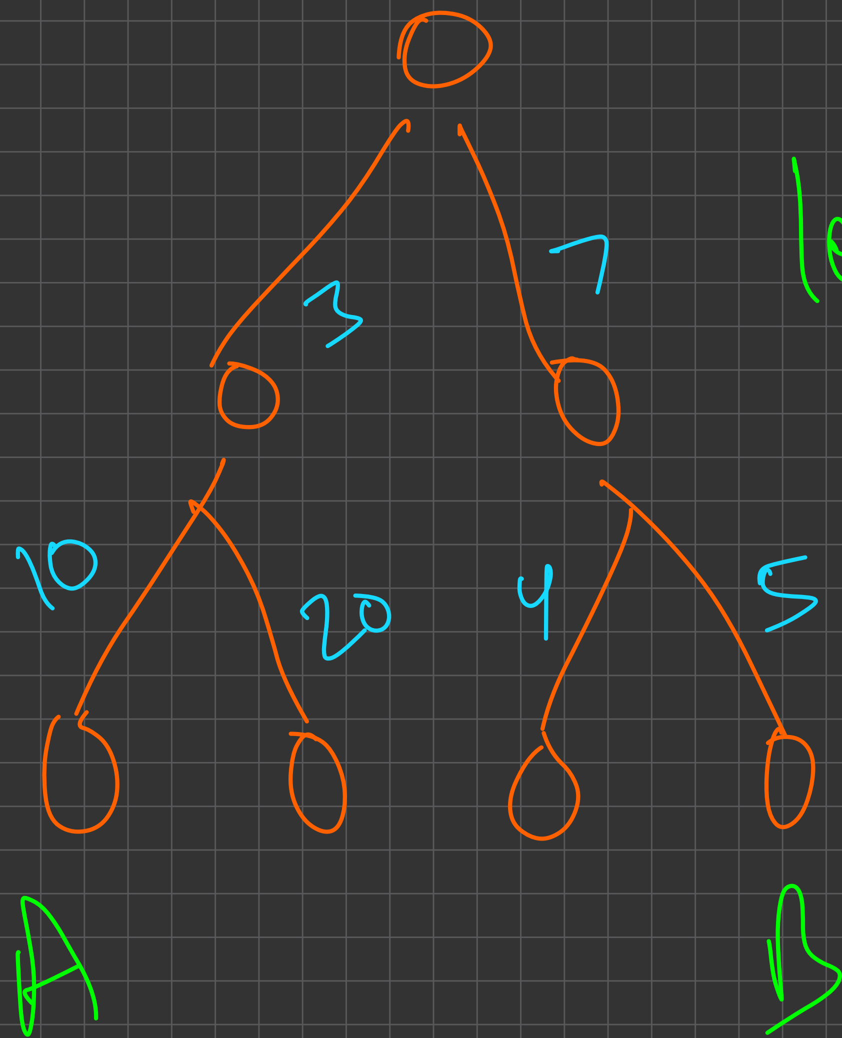
if there are any 2 nodes for which there doesn't exist a path b/w them

Directed vs Undirected



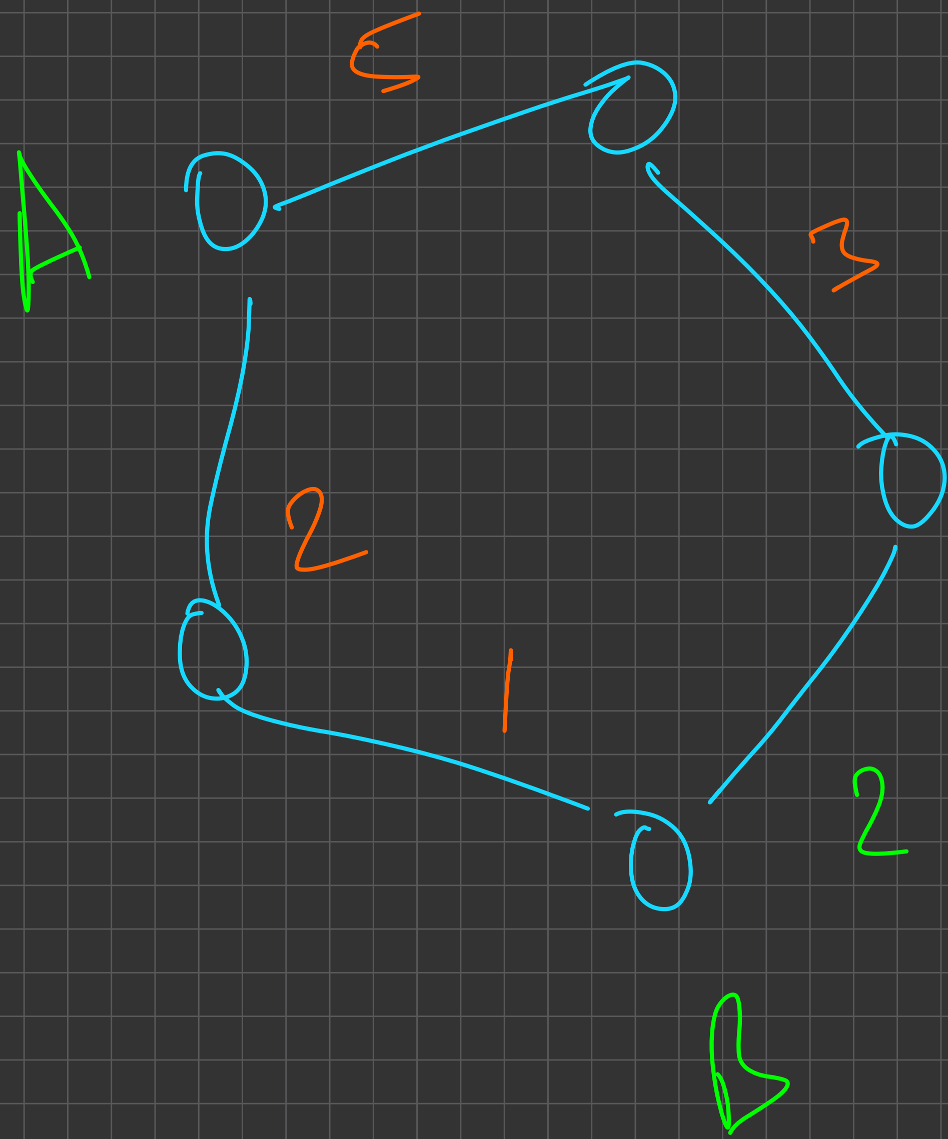


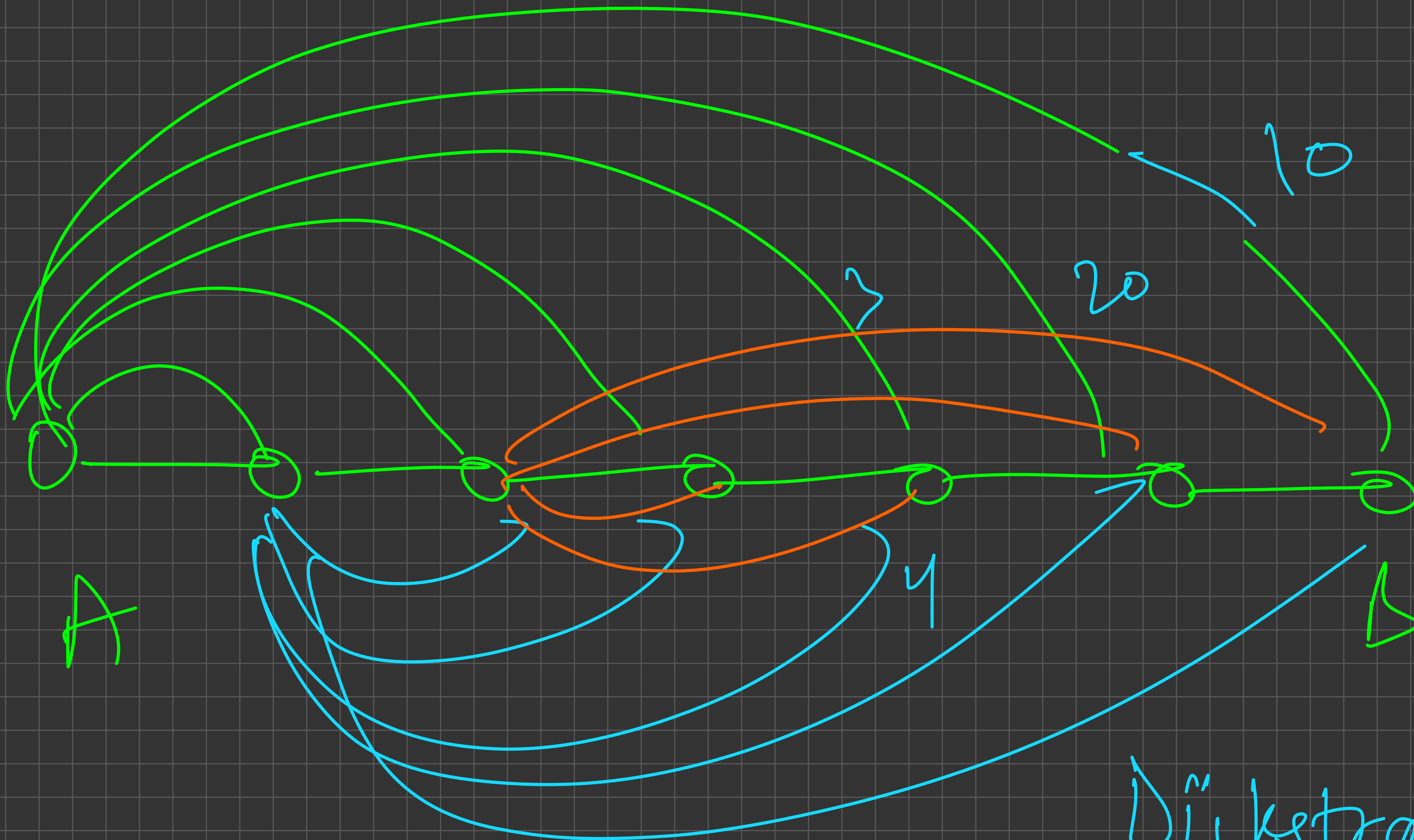




$$\text{level}(A) + \text{level}(B) - 2\cdot\text{level}(\text{lca})$$

$$\text{dist}(R, A) + \text{dist}(R, B) - 2\cdot\text{dist}(R, L)$$

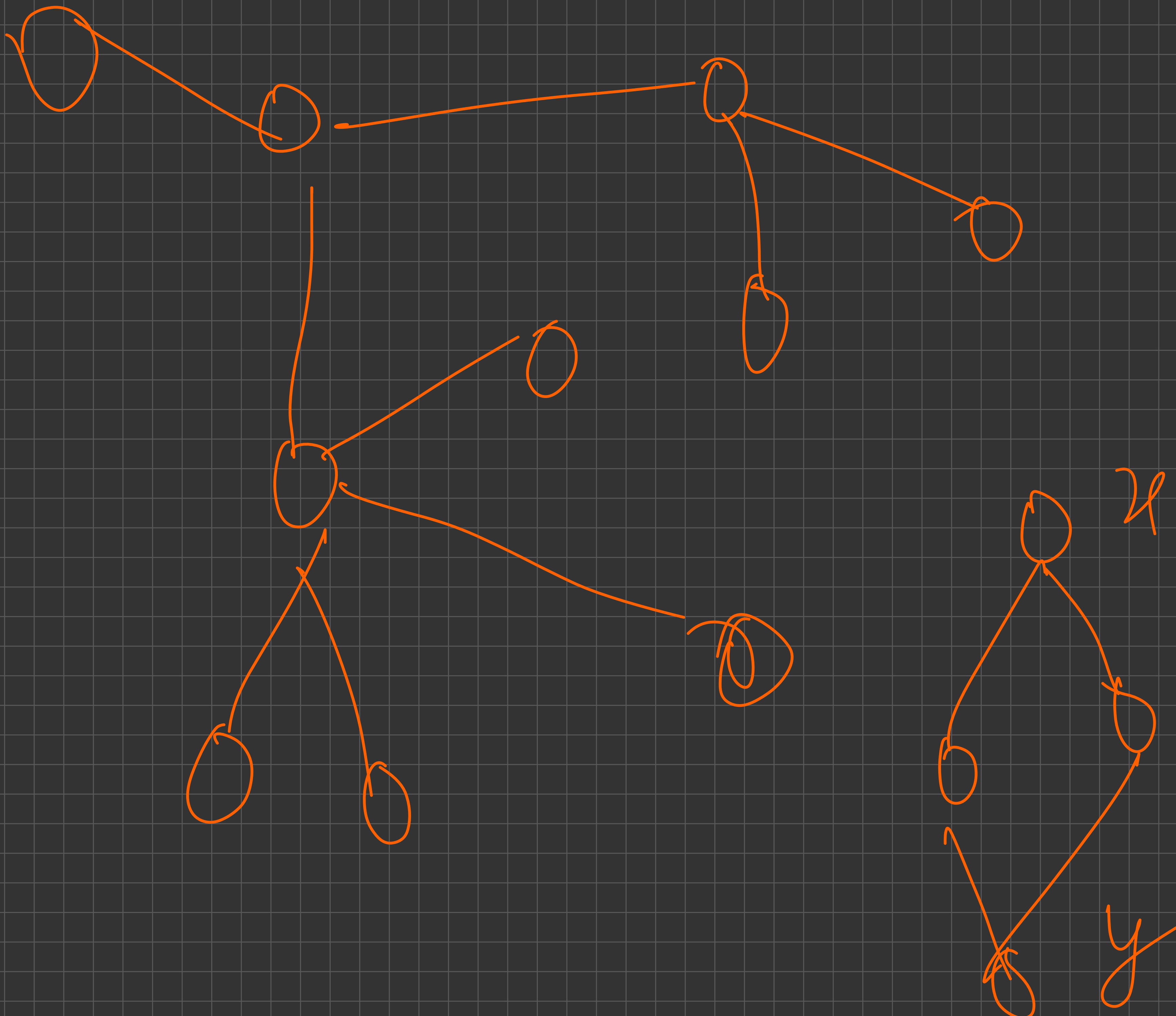

A
5
3
2
1
2
B

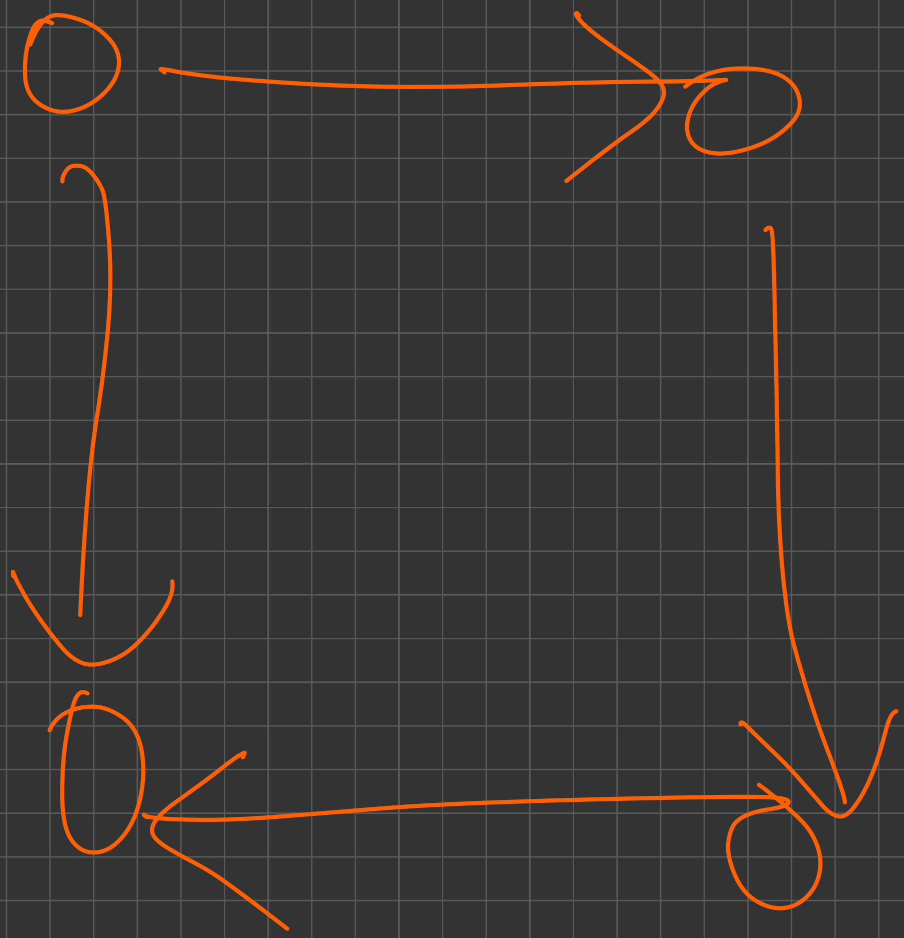


Dijkstra

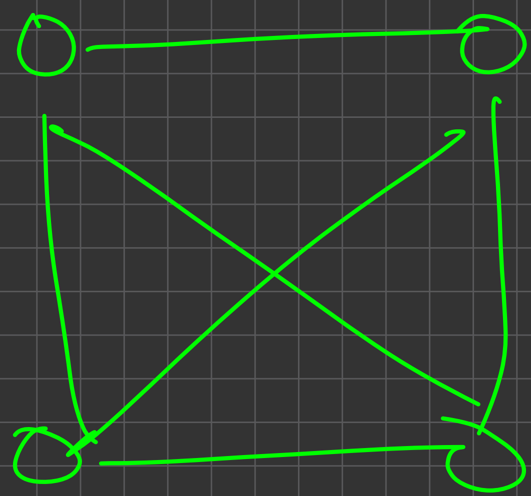
Floyd Warshall

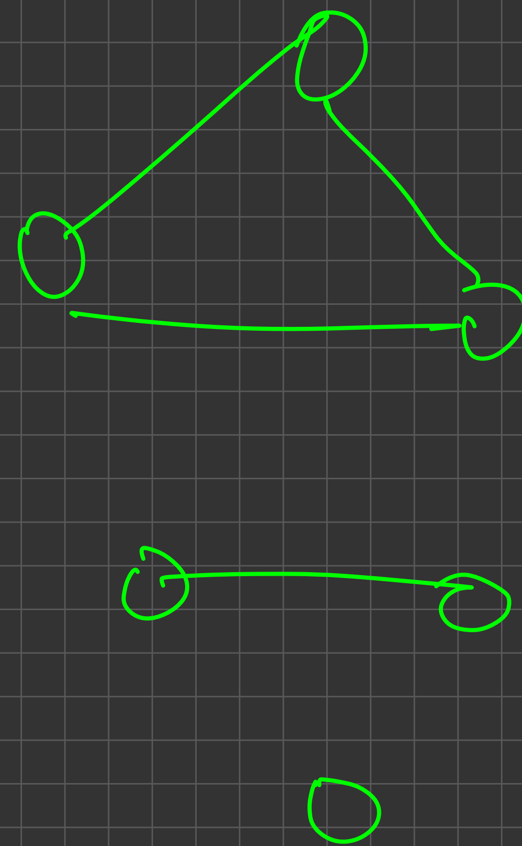
Bellman Ford

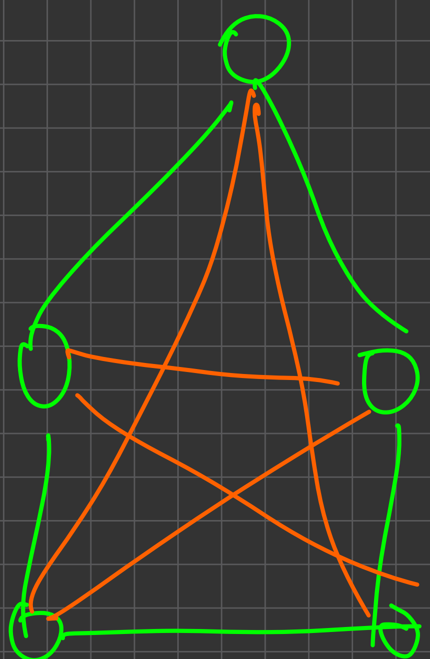

x
y



Complete Graph







n nodes

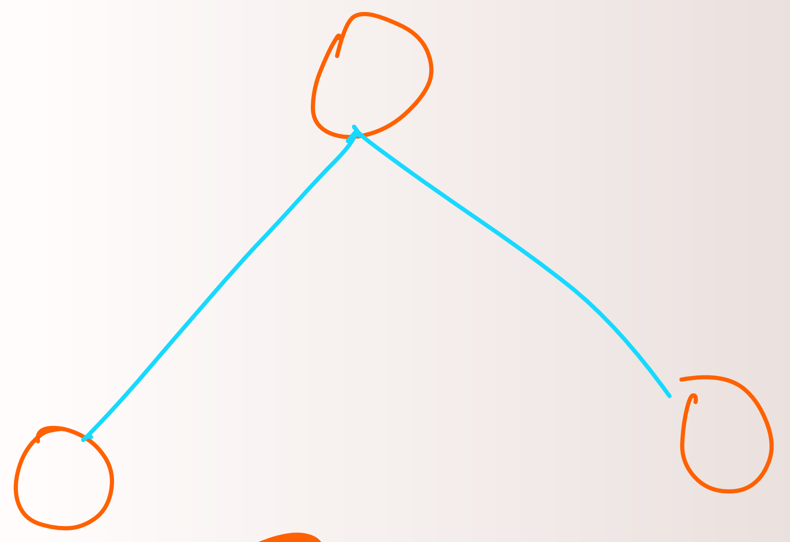
$^nC_2$ edges

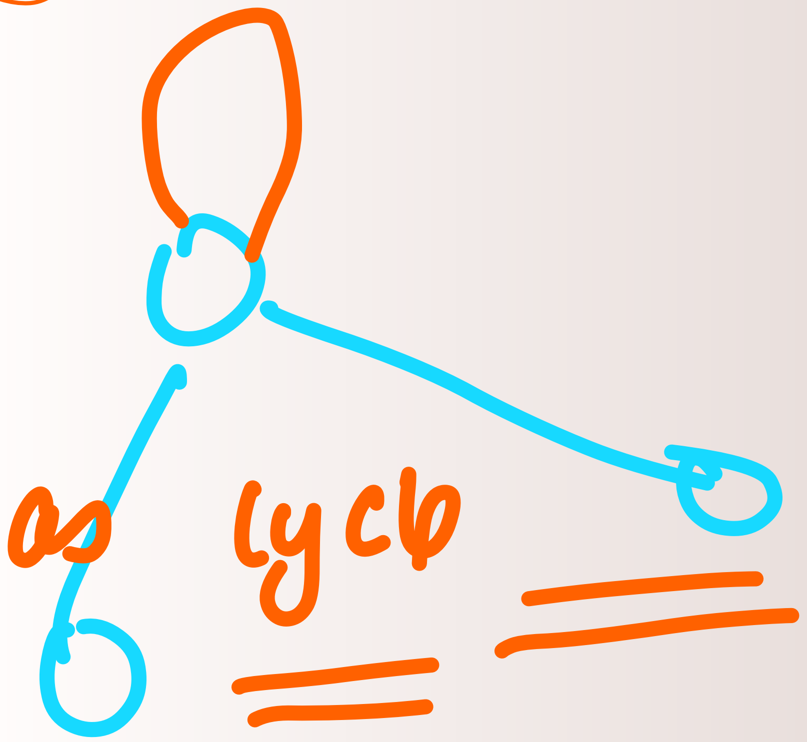
$$\frac{n\cdot(n-1)}{}$$

$$\left(n\cdot(n-1)\right)/2$$

Common Terms

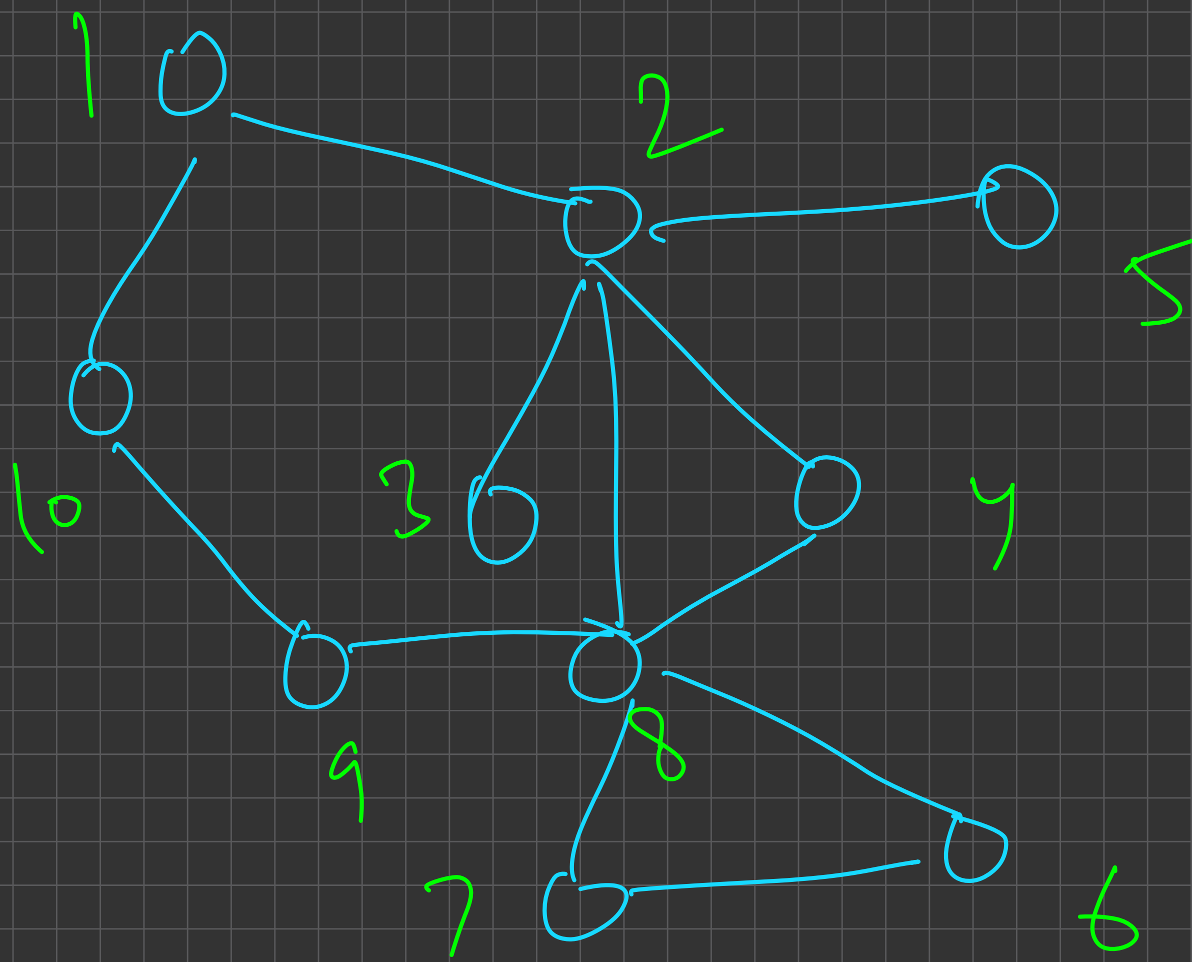
- Vertices + Edges
- Neighbours + Degree
- Self loop → not considered as cycle
- Path + Walk + Cycle
- Simple Graph
- Bridge + Articulation Point





$n \geq 2$

Path

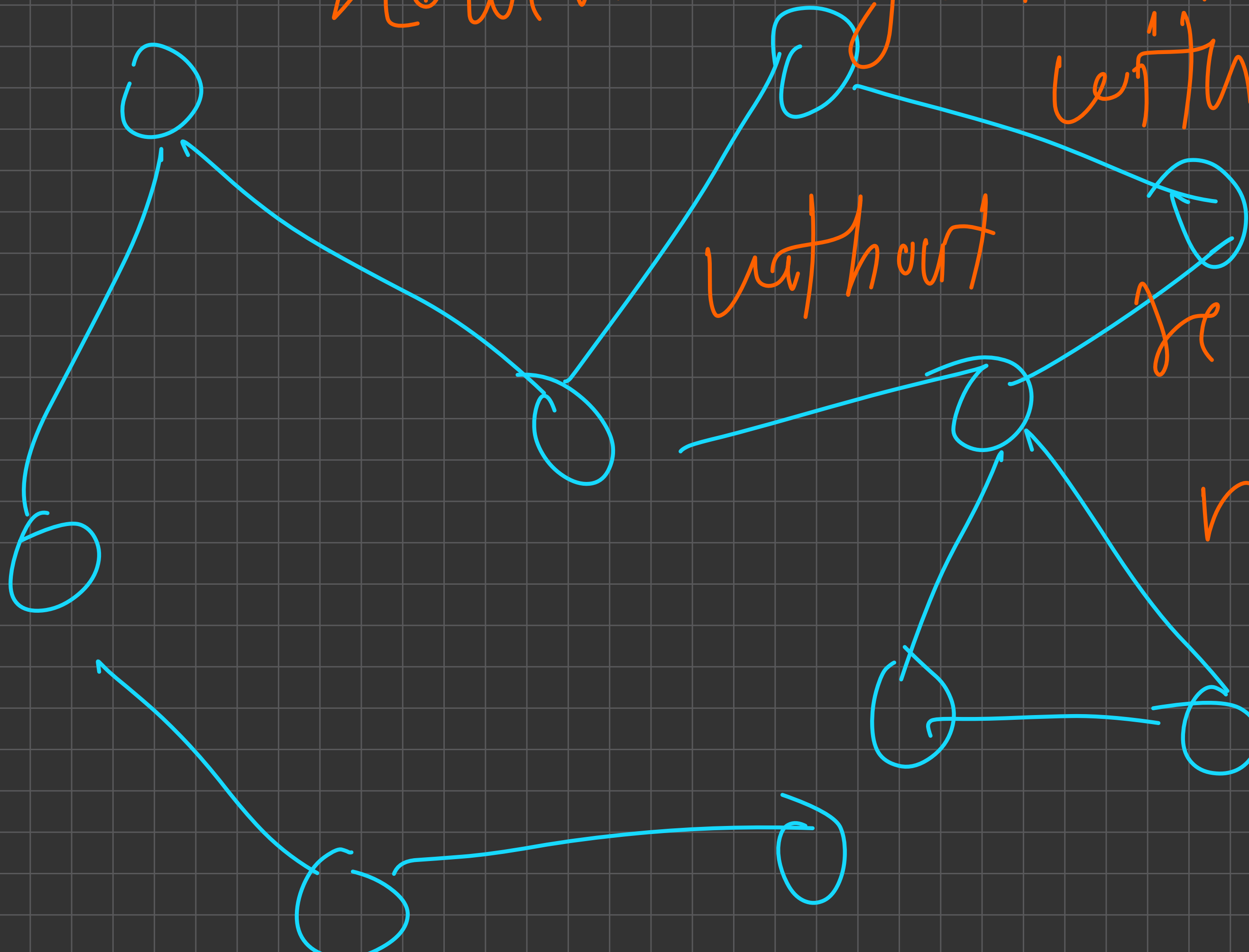


A to B

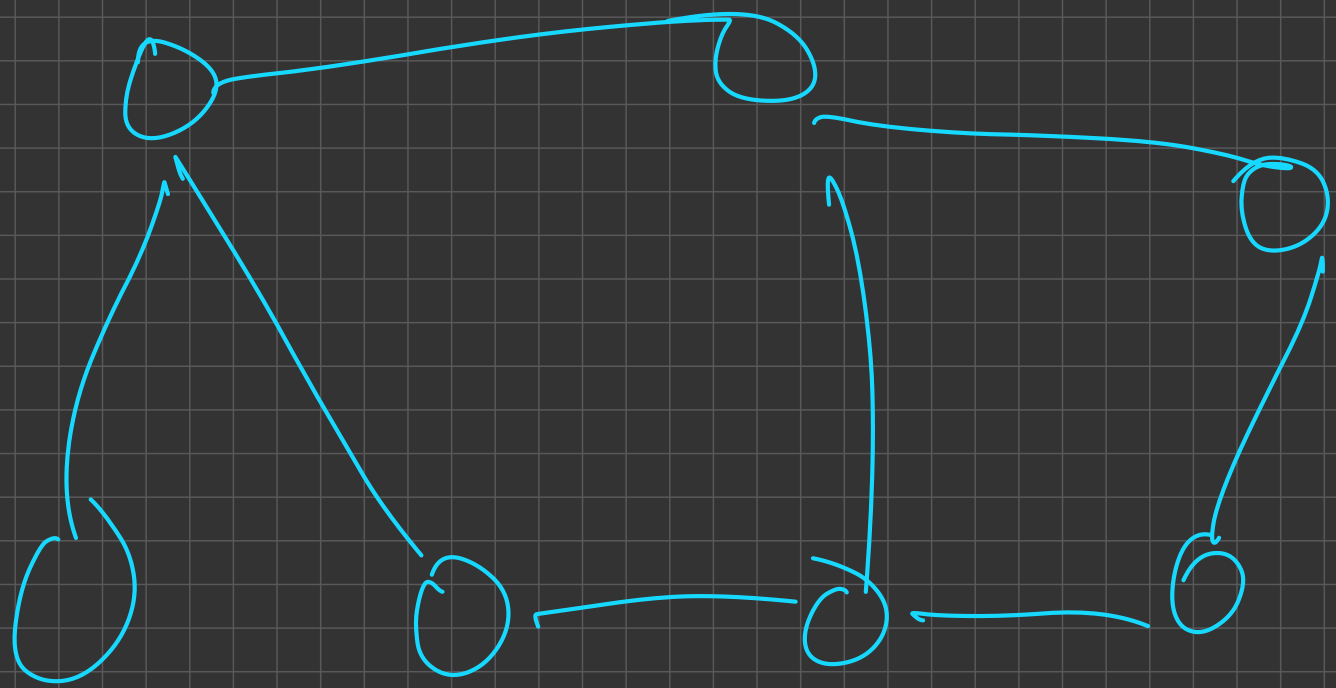
without repeating nodes and edges

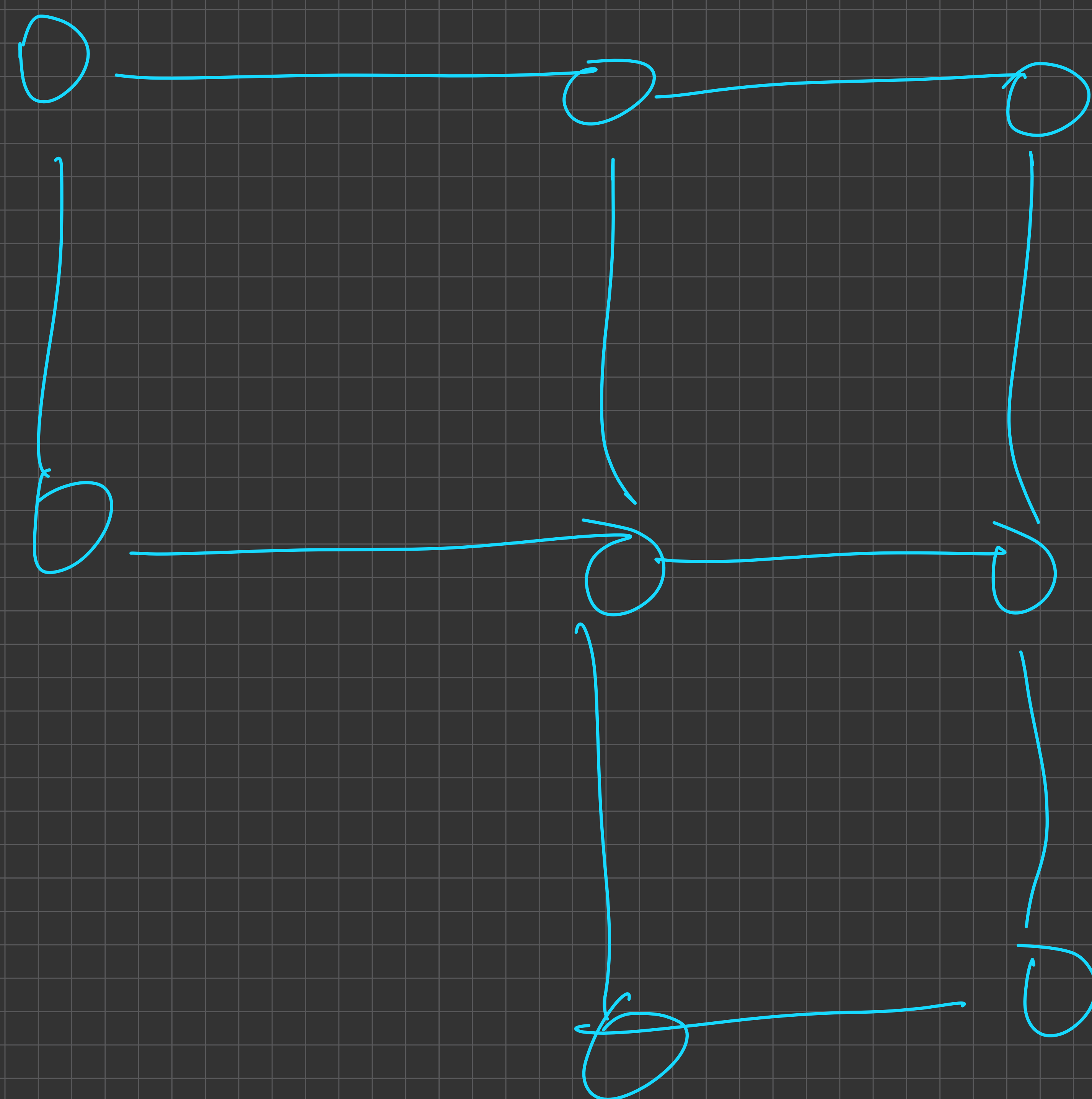
Walk

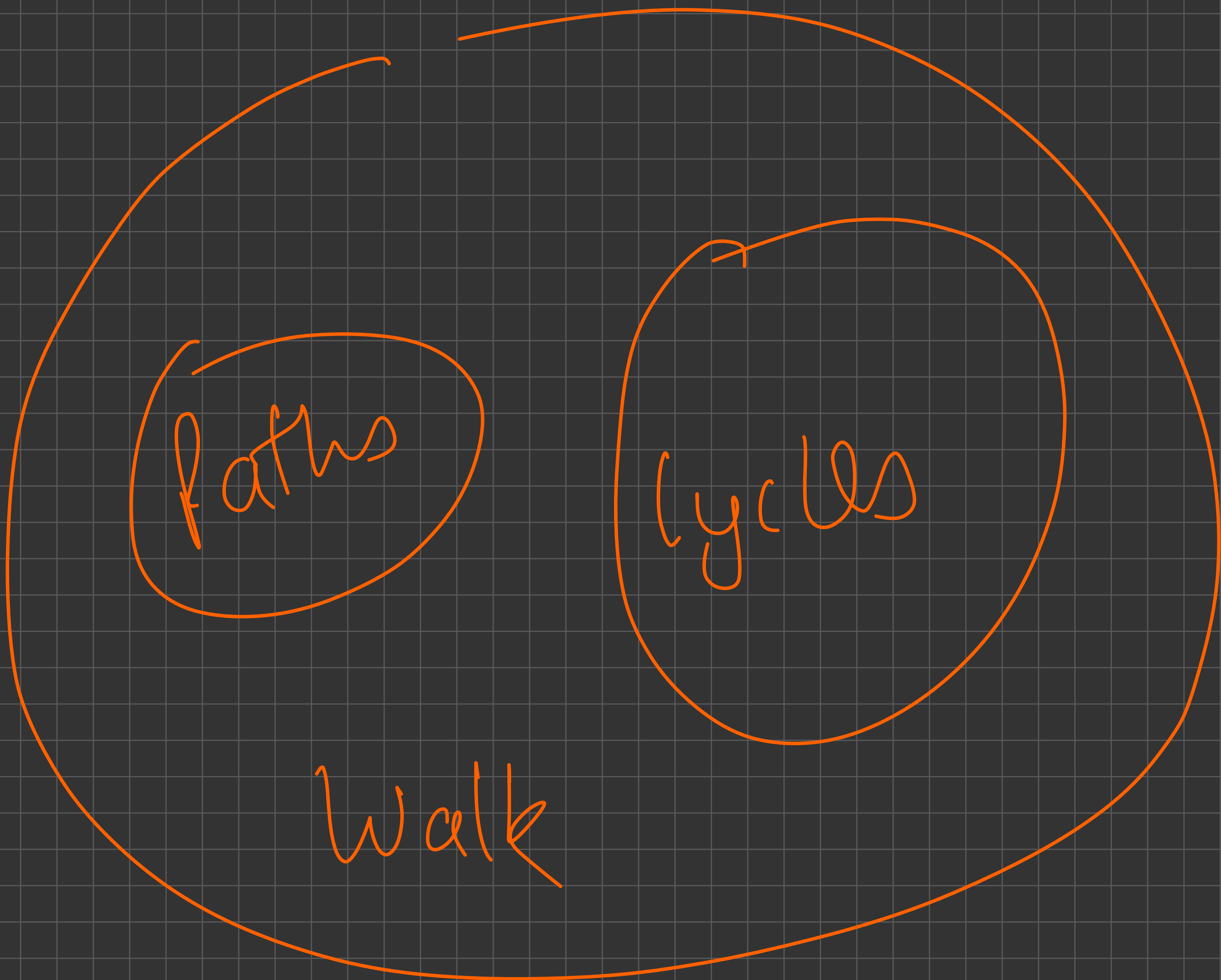
Go from A to B however you like with or without repeating node and edge



Cycle Go from A to A without repeating nodes & edges except the starting and ending node



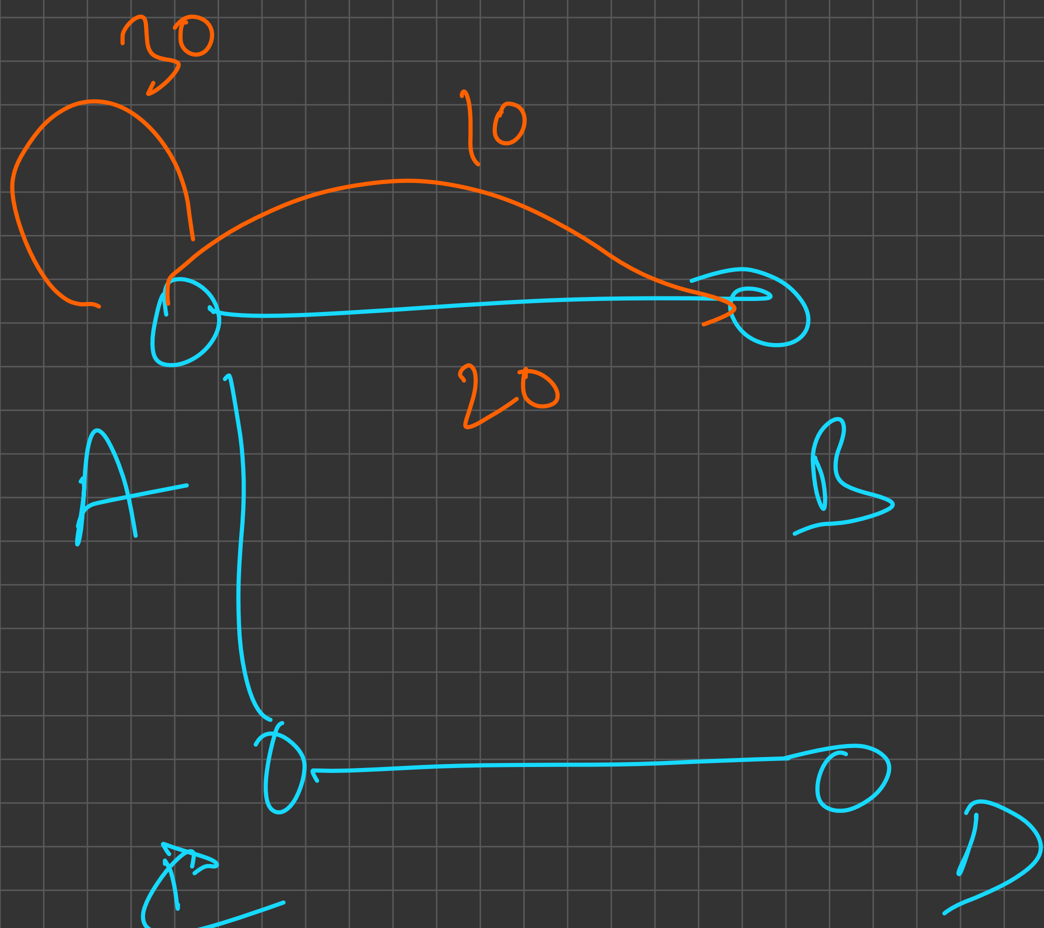




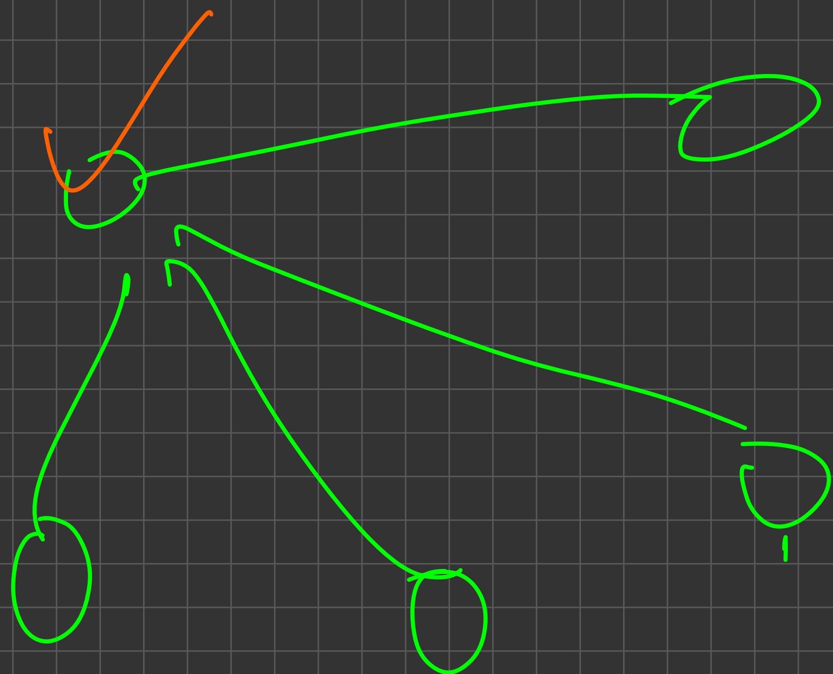
Paths
Cycles
Walk

Simple Graph

Graph with no self loops & multiple edges

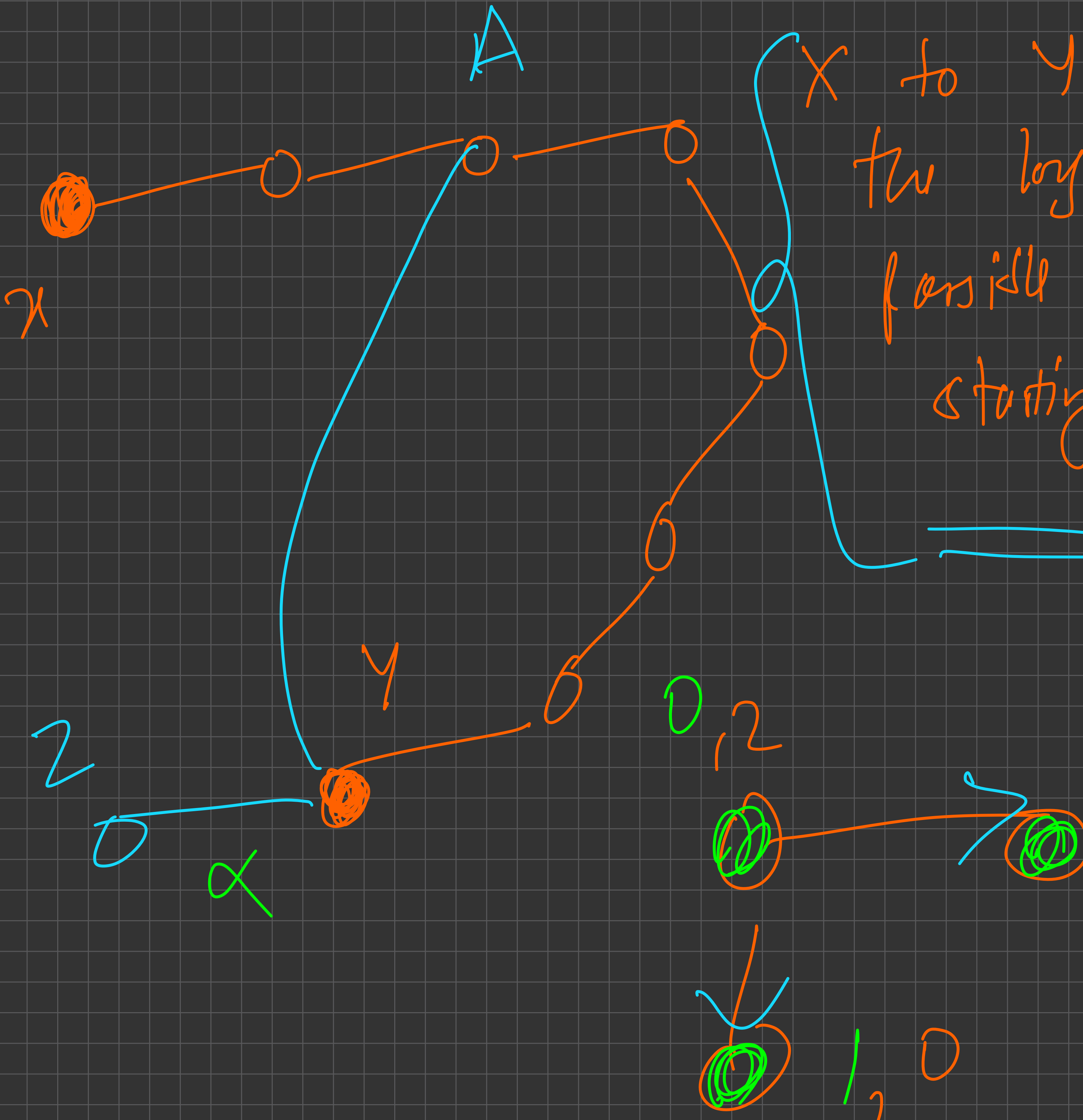


Bridges & Articulation Point



Some Common Results

- An undirected graph where each node has at degree at least 2 will contain a cycle
- A directed graph where each node has at least 1 in-degree and at least 1 out-degree will contain a cycle

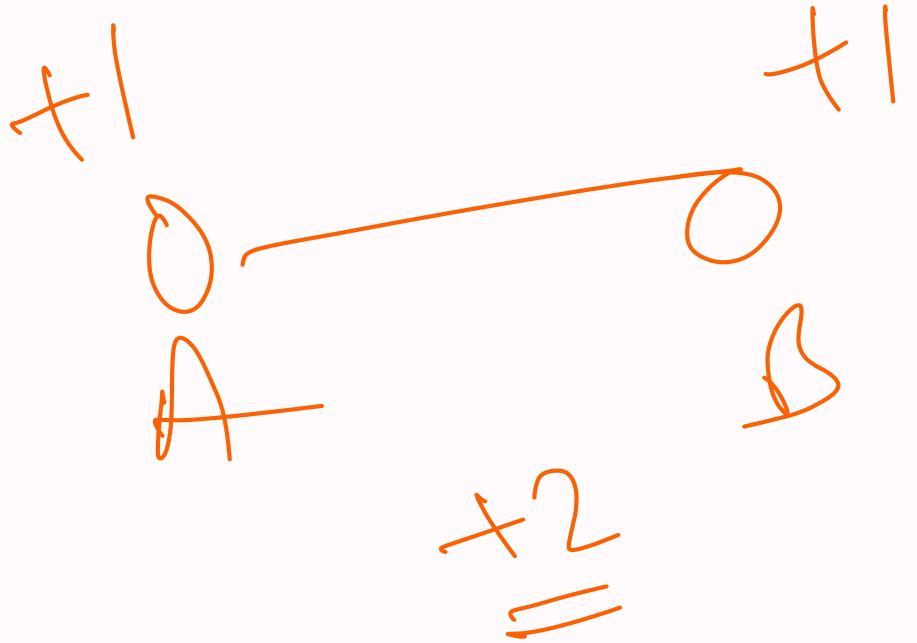


X to Y is the longest possible path starting from X

Some Common Results

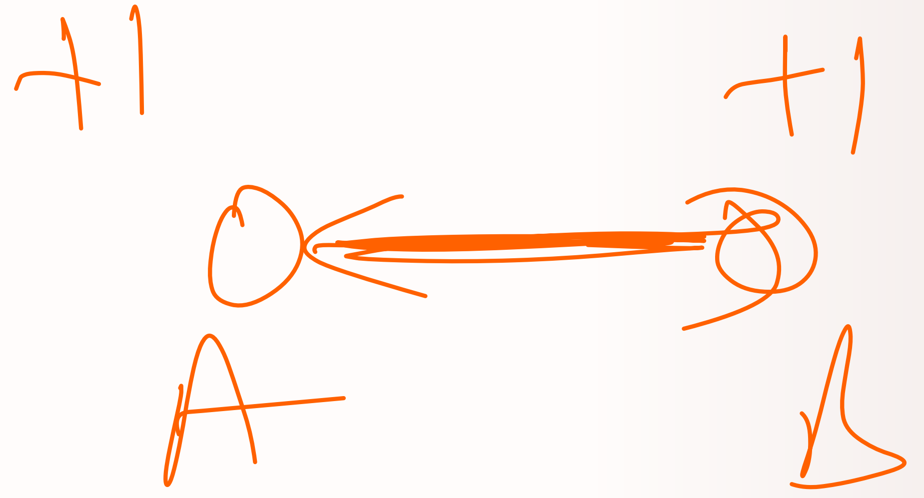
- The sum of all degrees is even. The number of vertices with odd degree is even.

- Some more as we move ahead…



x

$2 \cdot x$



n

$2 \cdot n$

Sum = even

$a_1 + a_2 + a_3 + a_4 + a_5$

1 3 5 2 4

$\underbrace{1 + 3}_{+7} \quad 5 = 2 \cdot n$

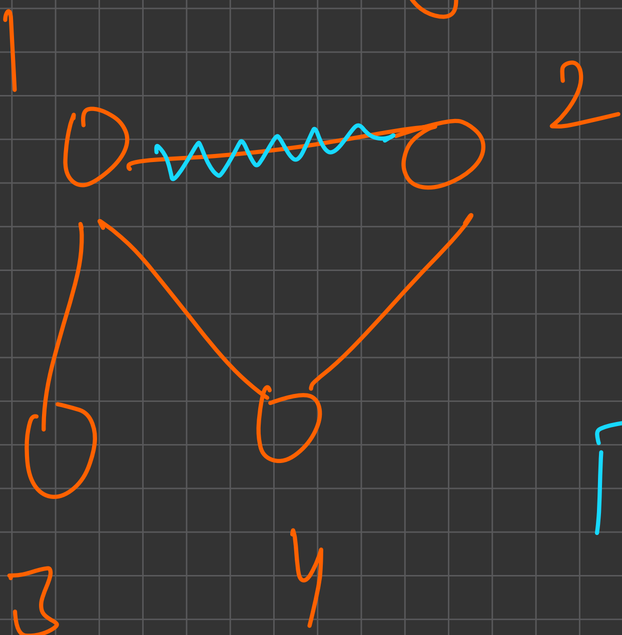
even

Representation

- Adjacency Matrix
- Adjacency List with Vector
- Adjacency List with Set
- Pros and Cons of each
- How is Input given in problems?

Adjacency Matrix

Space $O(n^2)$



	1	2	3	4
1	0	1	1	1
2	1	0	0	1
3	1	0	0	0
4	1	1	0	0

Updating, deleting, adding edges

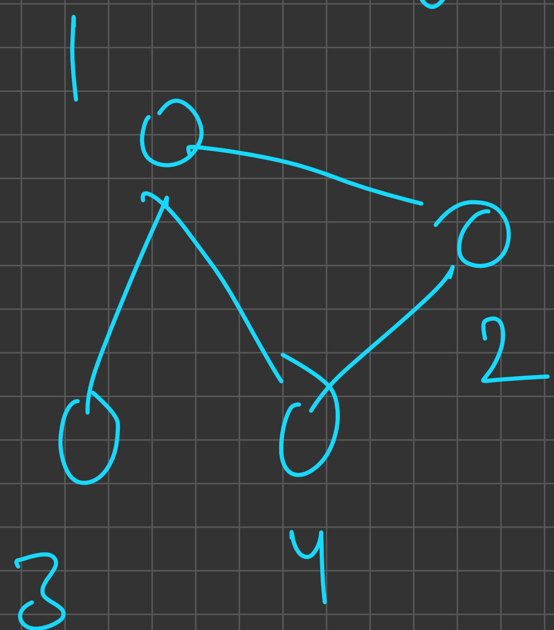
$O(1)$

traversals are heavy $O(n^2)$

n = nodes, m = edges

n^2

Adjacency list



Space $O(n+m)$

traversal → $O(n+m)$

1 → 2, 3, 4

2 → 1, 4   vector

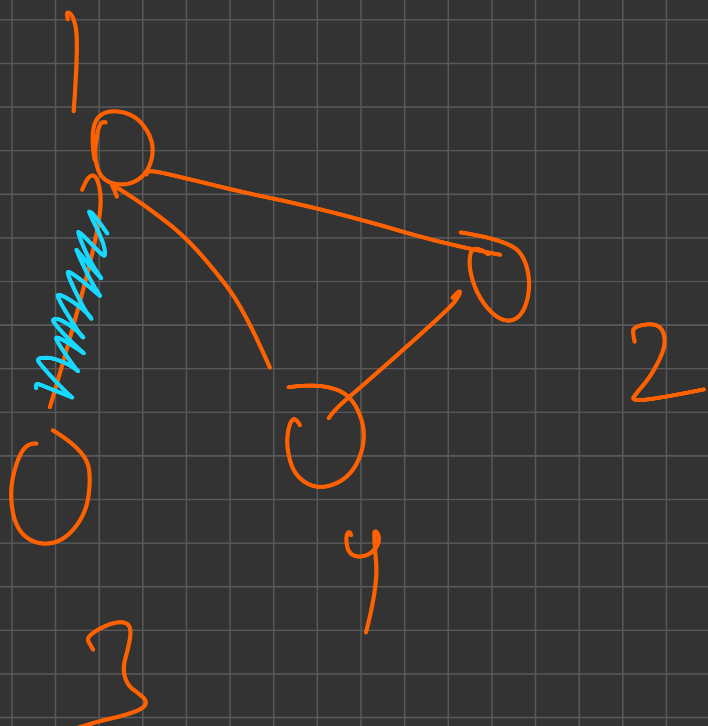
3 → 1

4 → 1, 2

Removing, updating → $O(n)$

Inserting → $O(1)$

Adjacency List with Sets



Set

1 → 2, 3, 4

2 → 1, 4

3 → 1

4 → 1, 2

$$\left.\begin{array}{l}\text{edges}[1].\text{erase}(3)\\ \text{edges}[3].\text{erase}(1)\end{array}\right\}\ O(\log n)$$

n nodes , m edges

$m \underset{\smile}{\sim} n^2$

A matrix

B list

C list with set

adding edges ; n^2

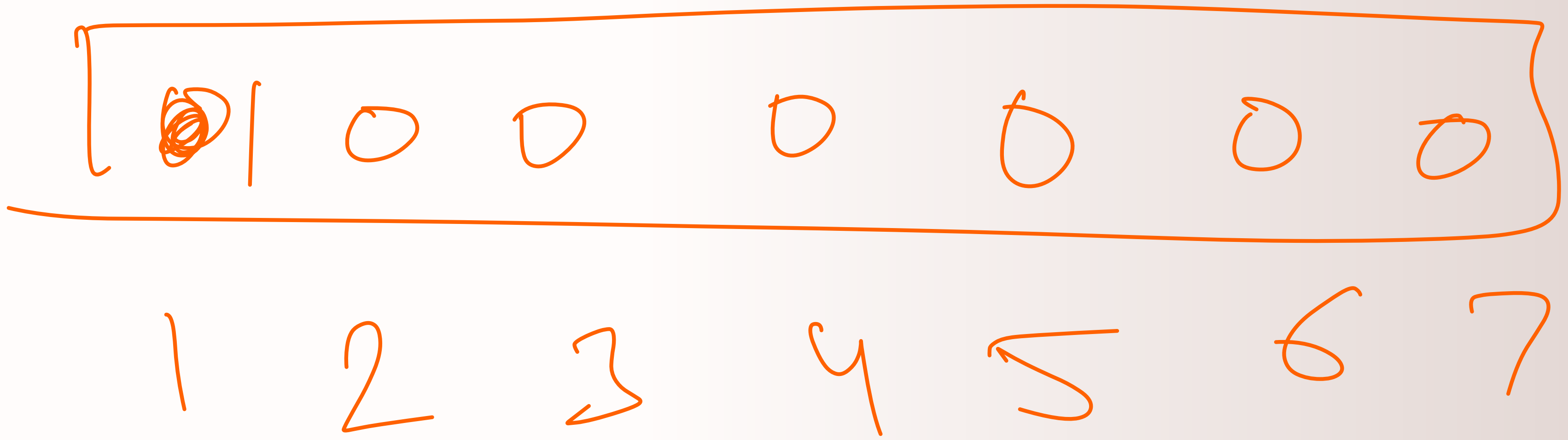
traversals

remove edges $O(1)$

```
void dfs (int cur, edges, parent)

    for ( child : edges (curr) )
```

Traversals

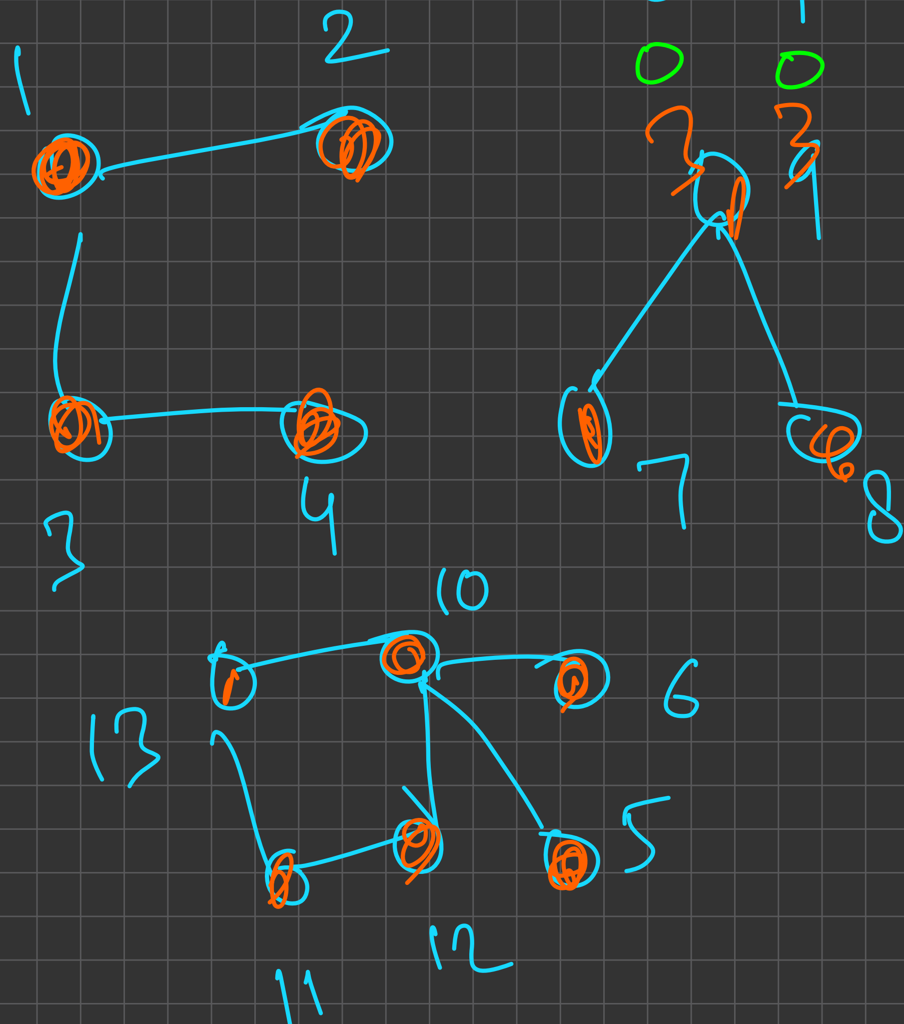


- DFS
- BFS (Single source and Multi source)
- Application of Traversals
  - Connected components (Problem)
  - Path construction
  - Cycle detection
  - Shortest Path (Problem)

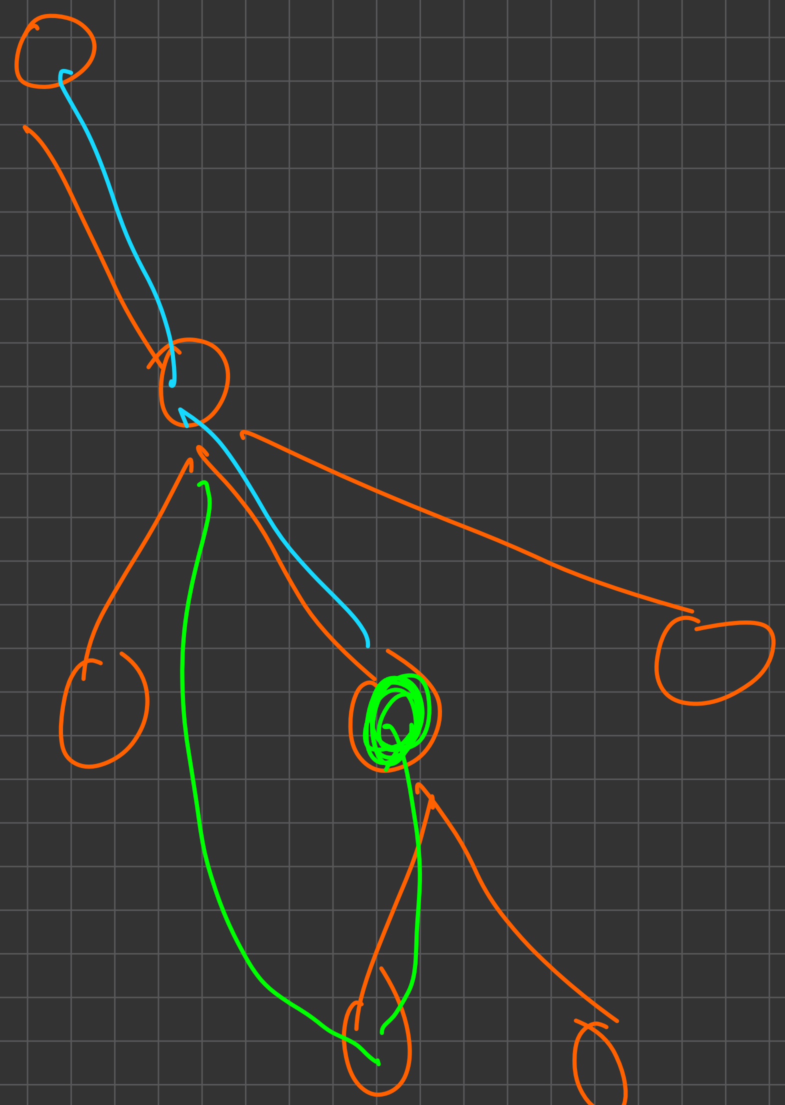
Connected Components :

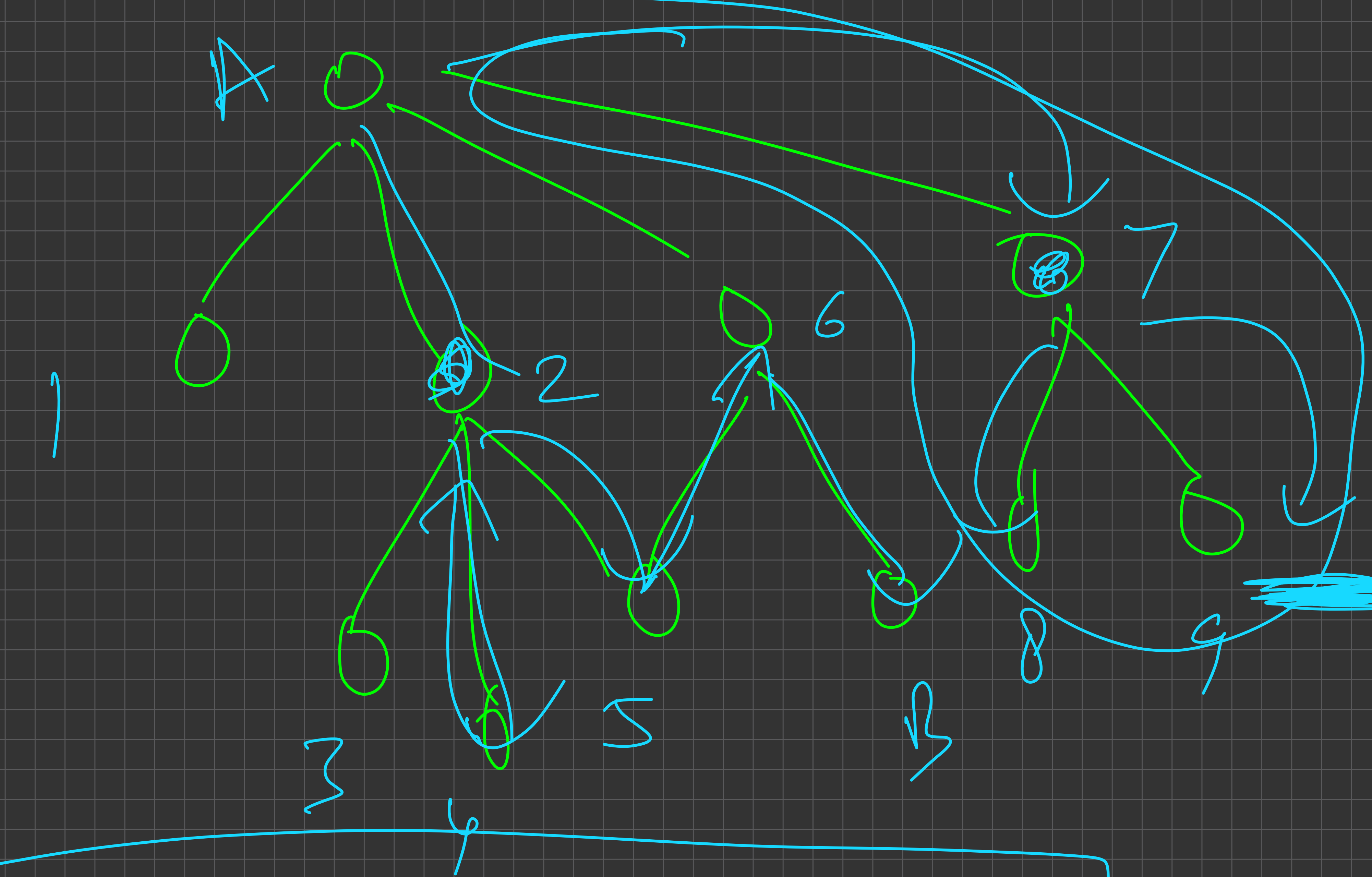
1	2	3	4	5	6	7
1 0	0 1	0 1	0 1	0 2	0 2	0

8	9	10	11	12	13
0 3	0 3	0 2	0 2	0 2	0 2

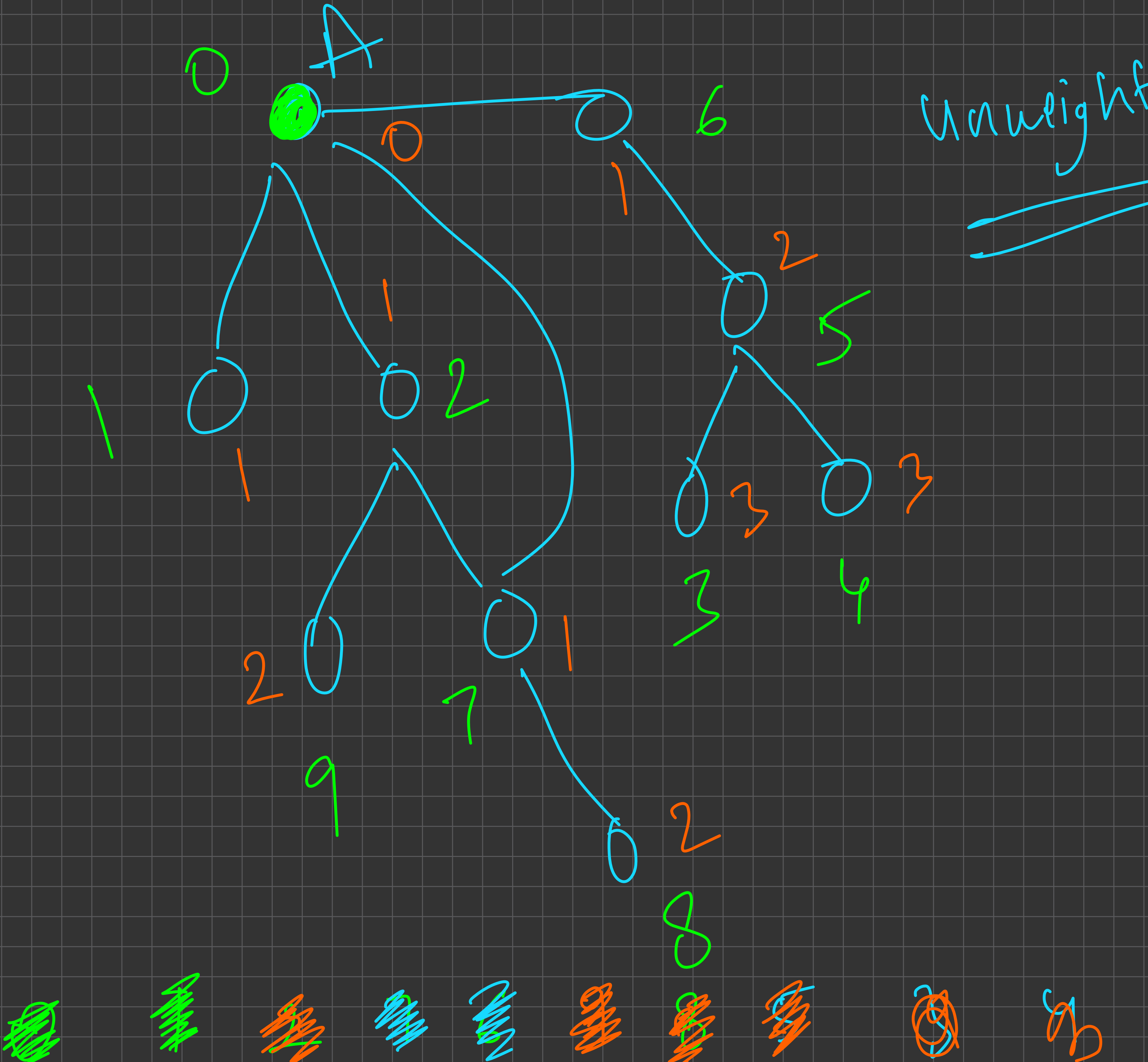


$\underline{O(n+m)}$

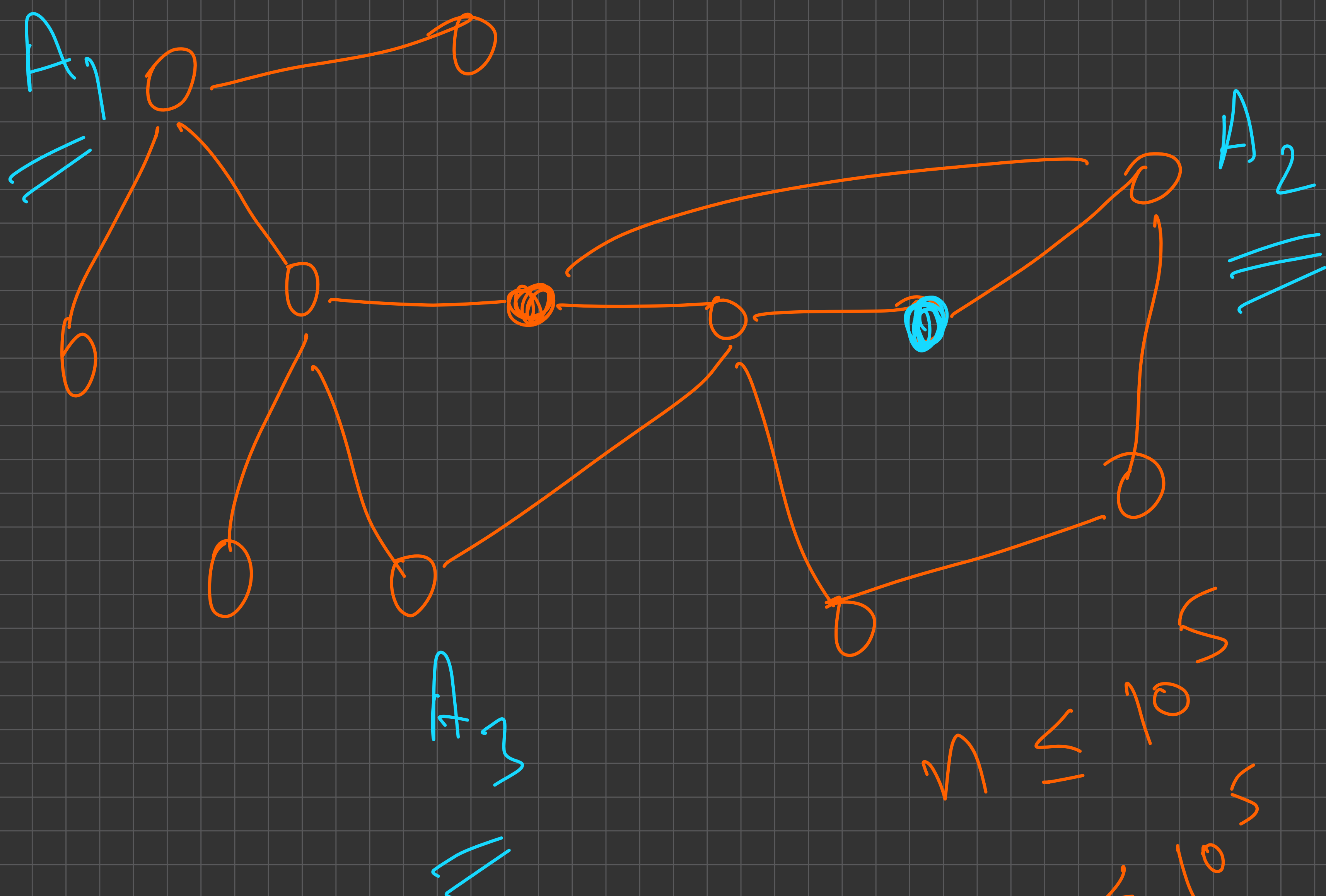




A 2 5 6 B



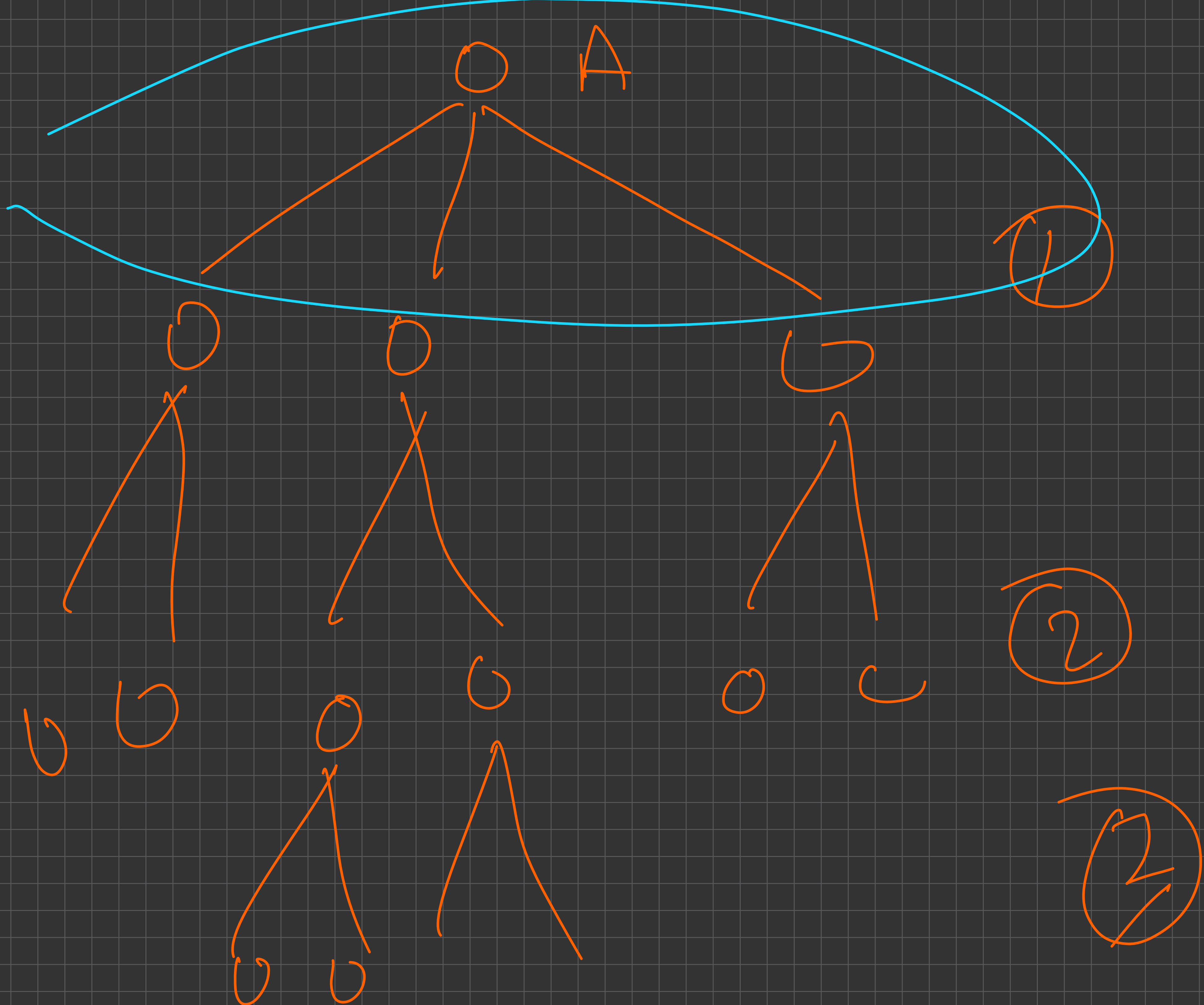
Unweighted

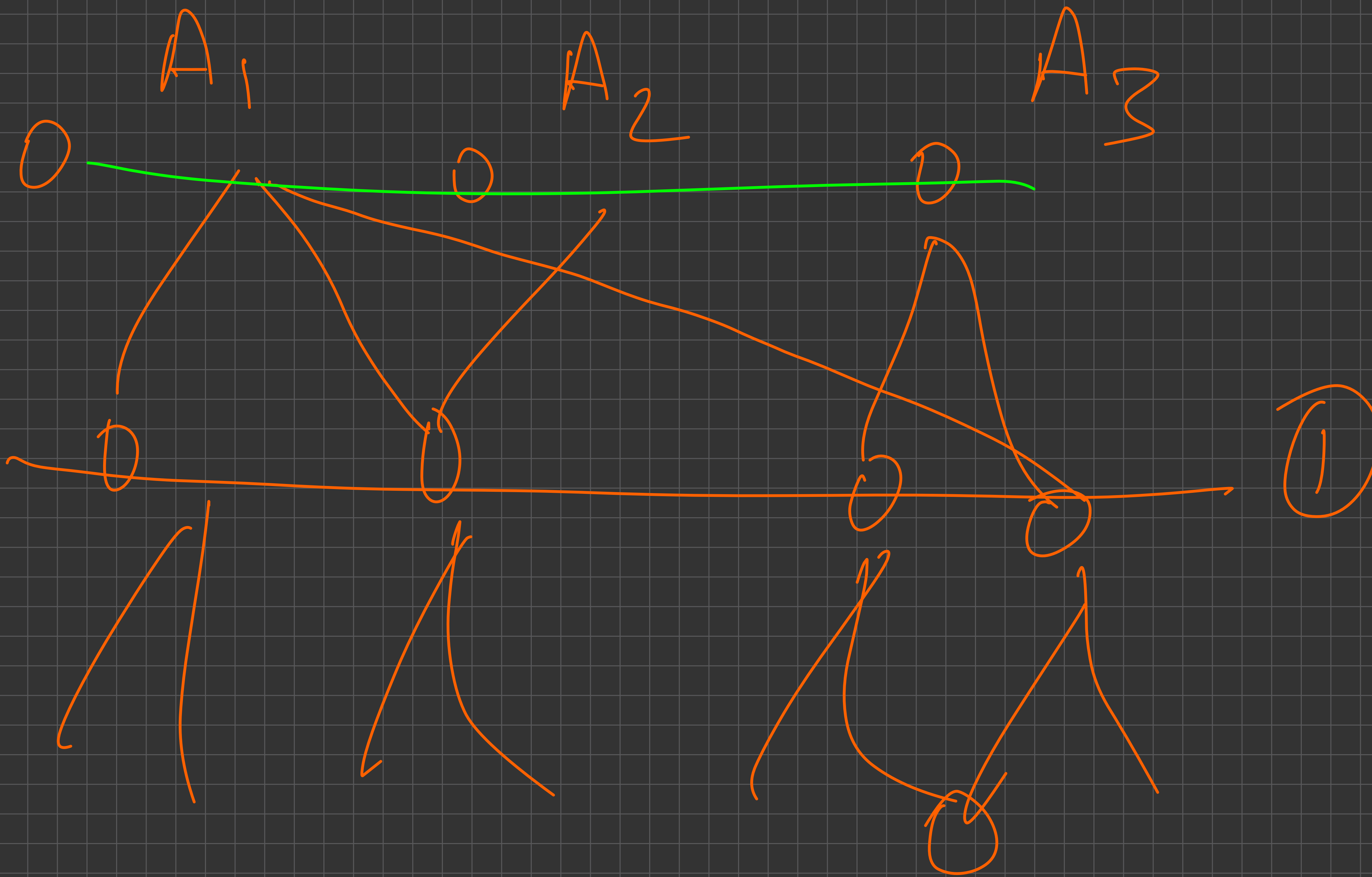


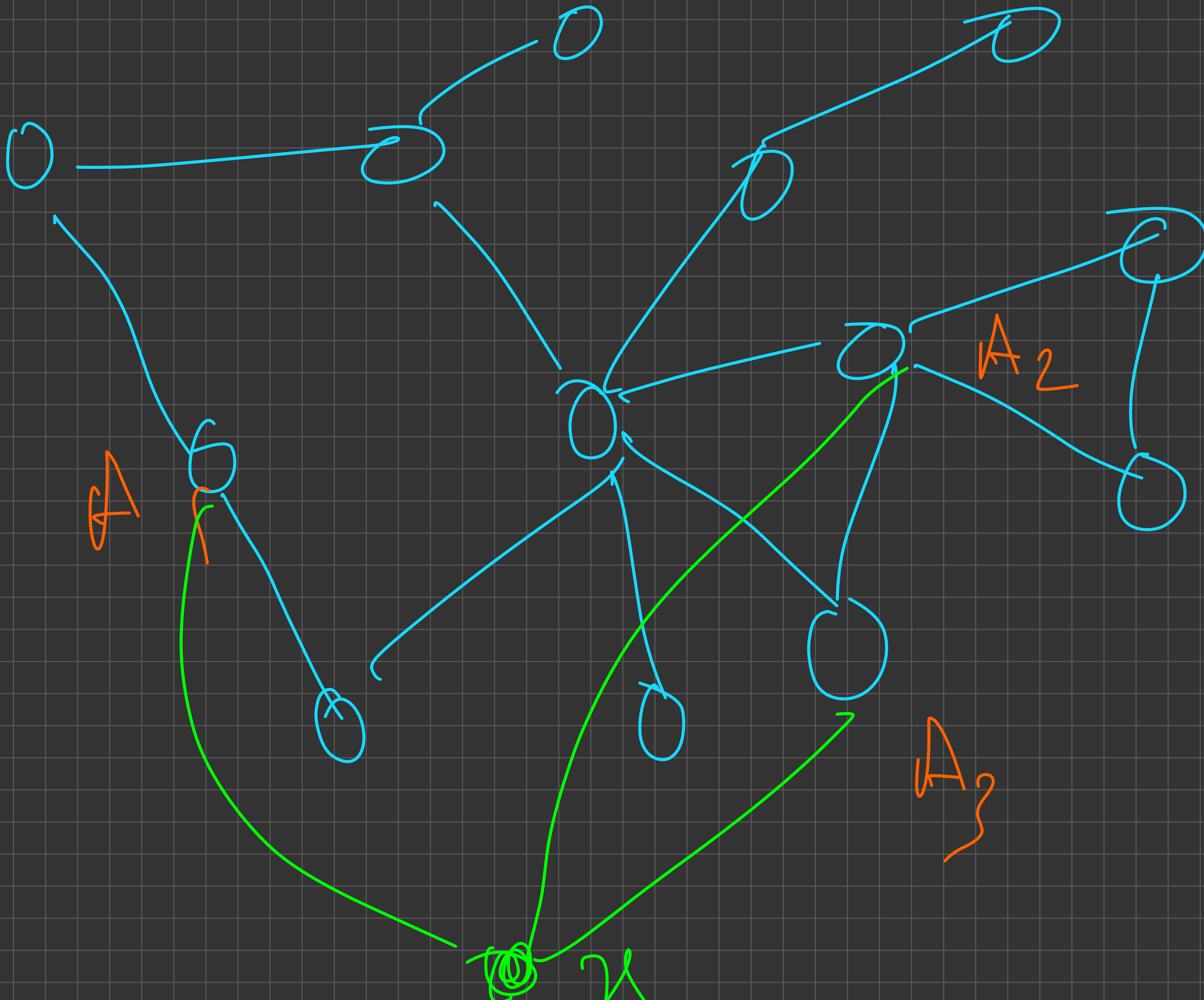
$n \le 10^5$

$m \le 10^5$

$k \le n$


A
1
2


A1
A2
A3
0
0
0
0
0
0
0
0
0
0


A
A2
A3
u

Bipartite Graphs

- Algorithm
- Common Properties
  - Odd Length Cycles
  - A Tree is always bipartite
- <u>Problem</u>

Tree