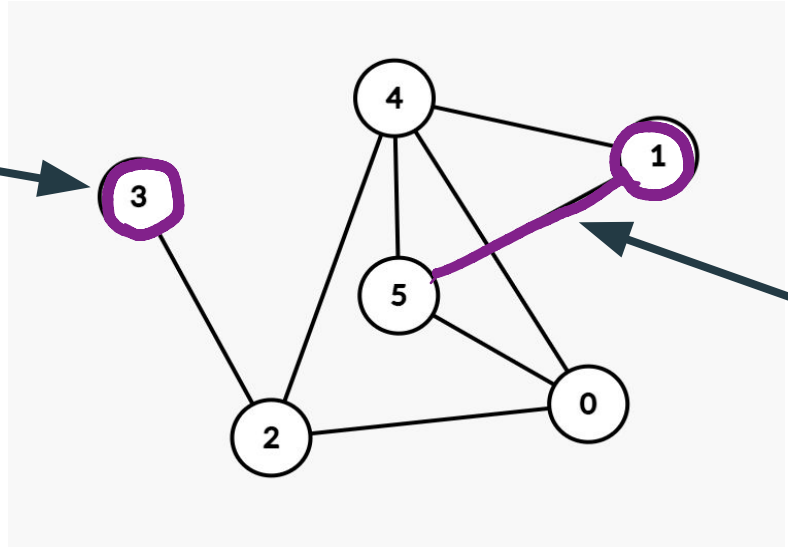
Introduction to Graphs

- Gaurish Baliga

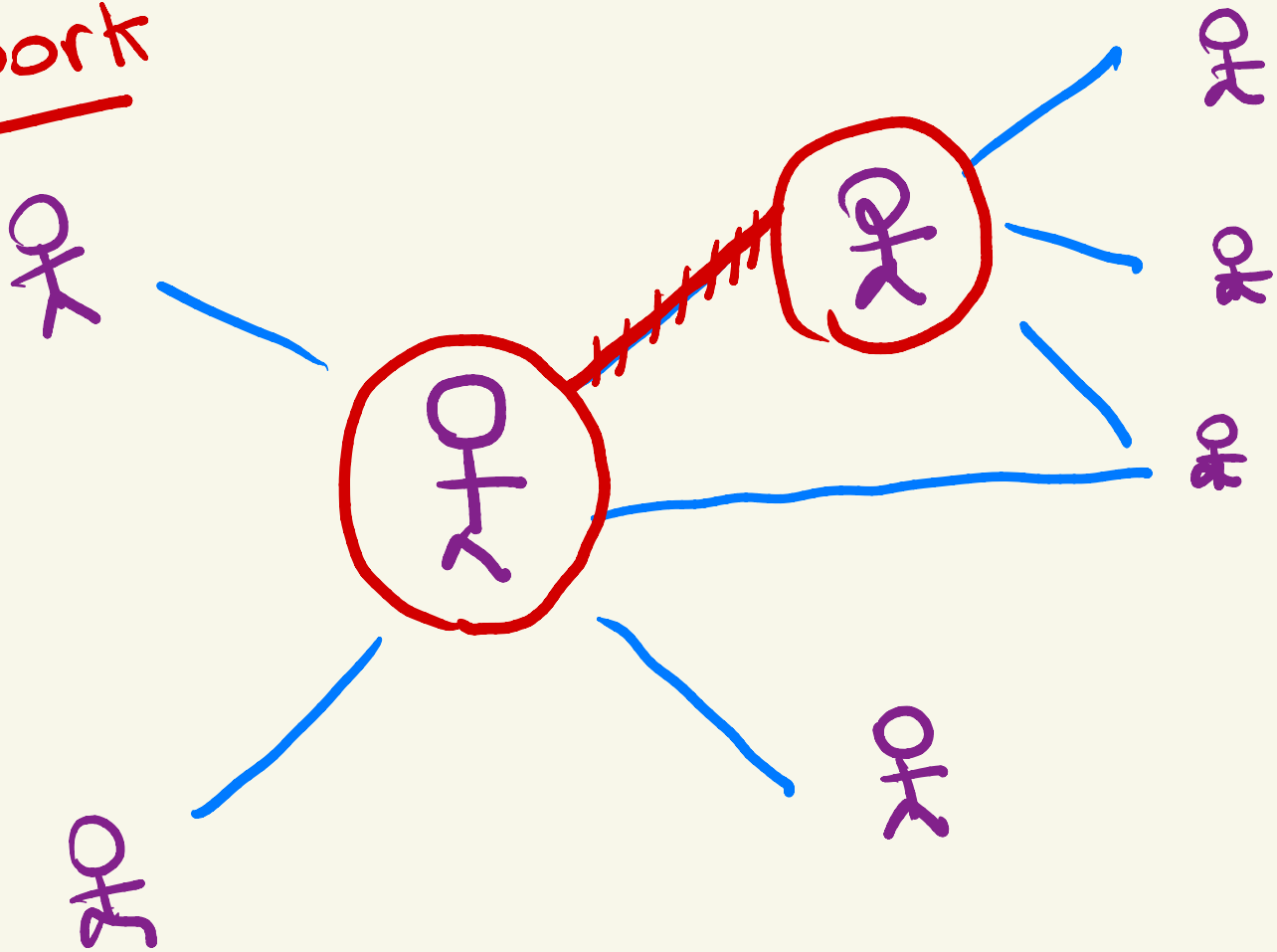
Goal

- Introduction to Graphs ✓
- Types of Graphs ←
- Common Terminologies ✓
- Representation of Graphs ✓
- Tradeoff Analysis and Common Results ✓

What is a Graph?



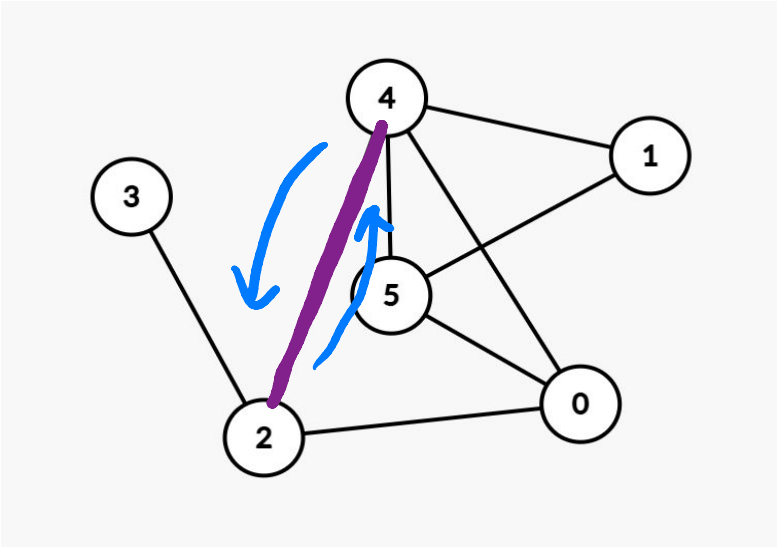
Network



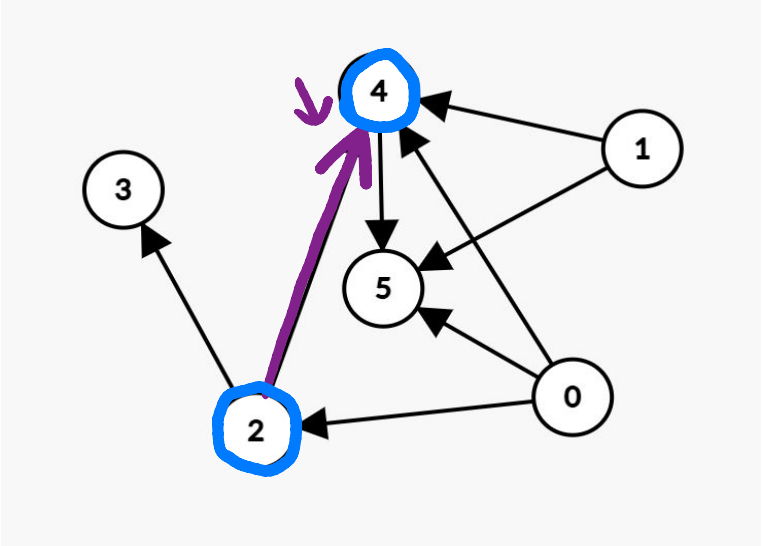
Types of Graphs

- Directed vs Undirected
- Weighted vs Unweighted
- Cyclic vs Acyclic
- Connected vs Disconnected
- Complete Graph

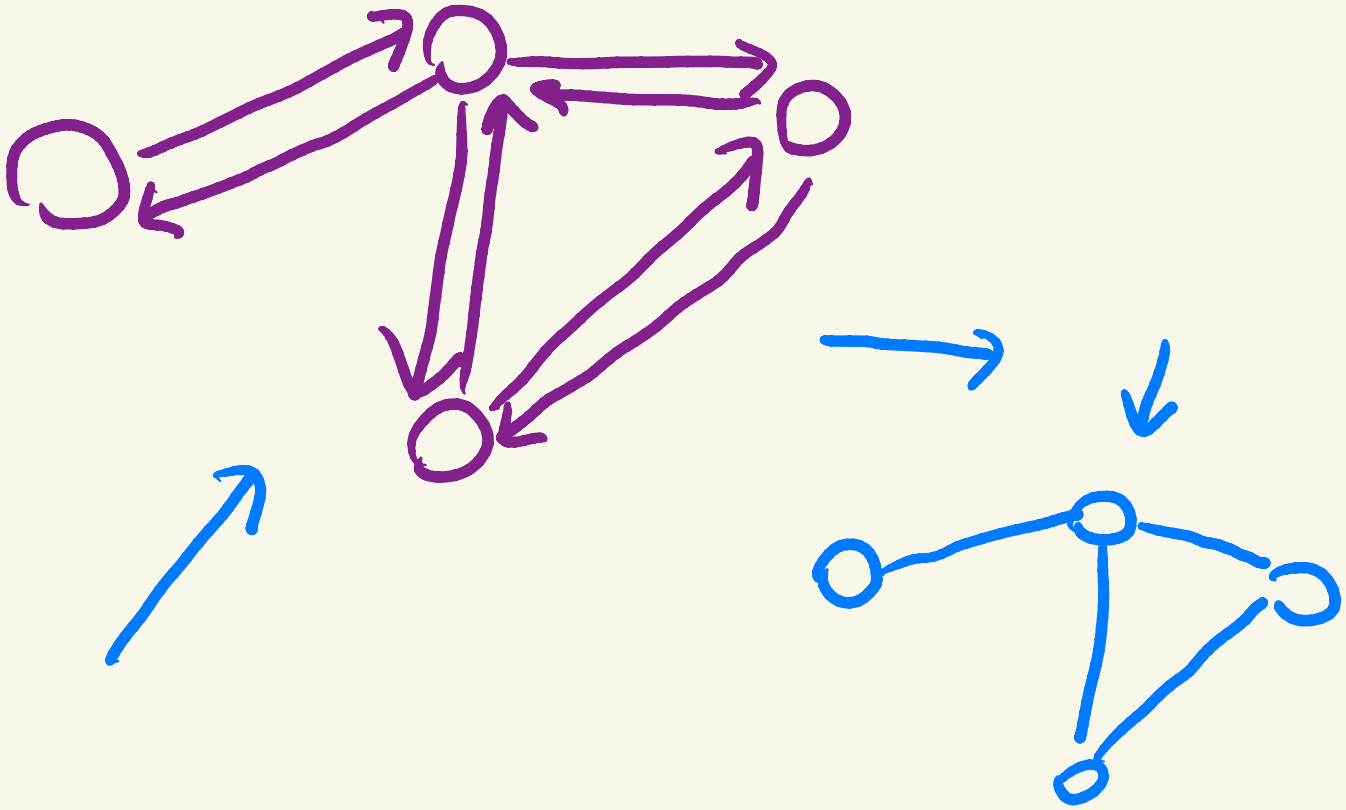
Directed and Undirected Graphs



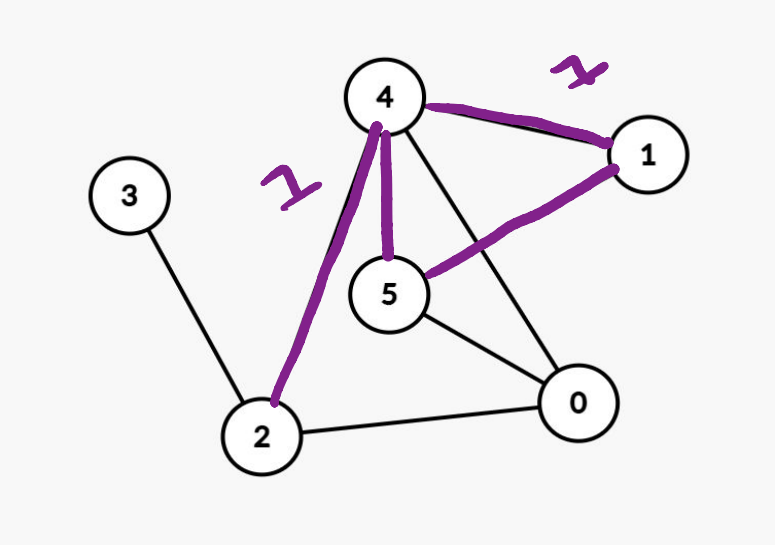
Undirected Graph



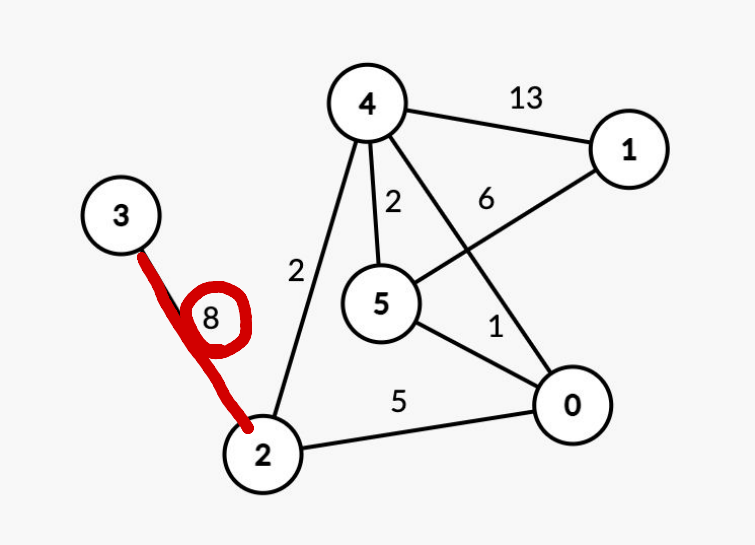
Directed Graph



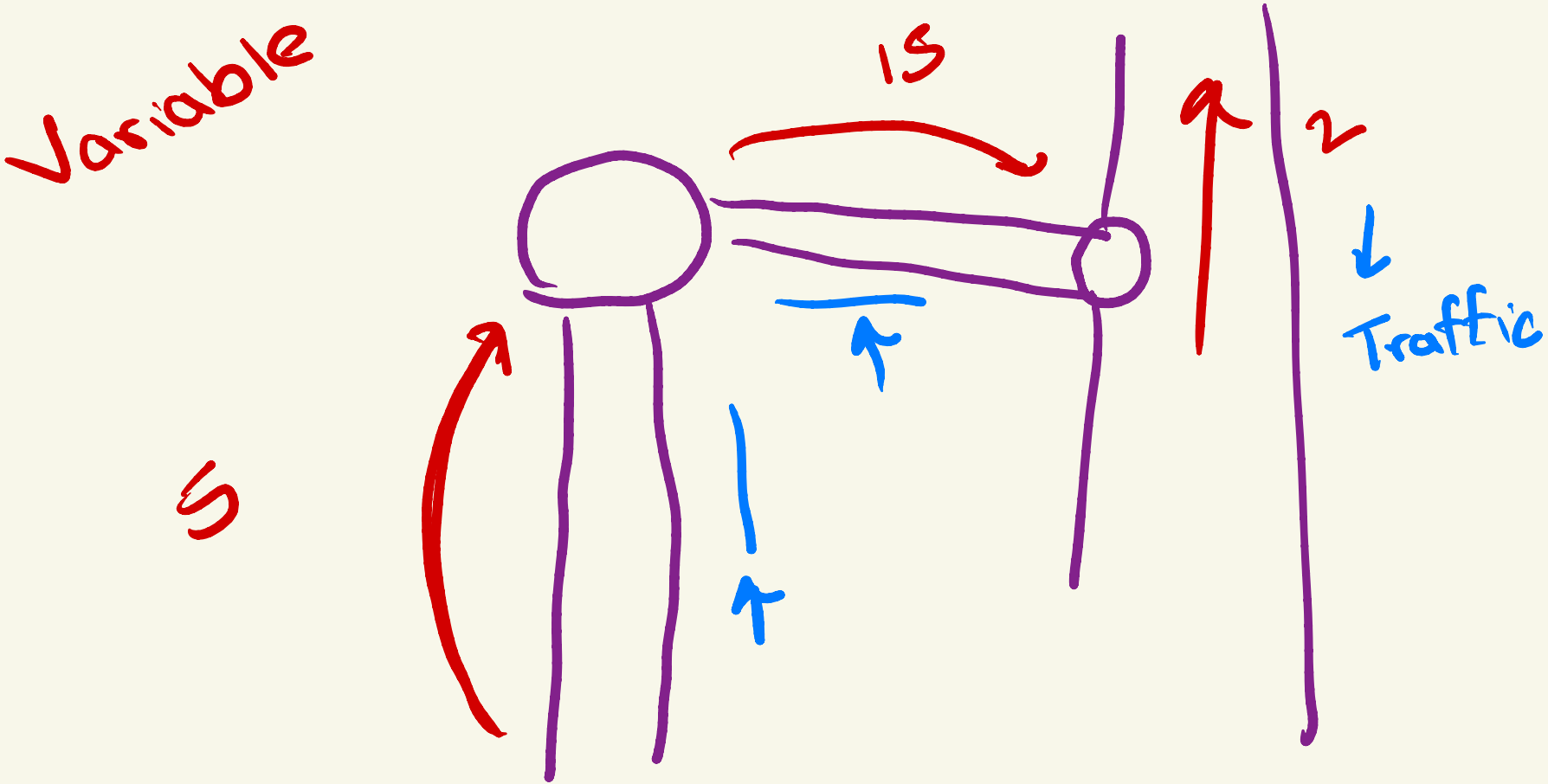
Weighted and Unweighted Graphs



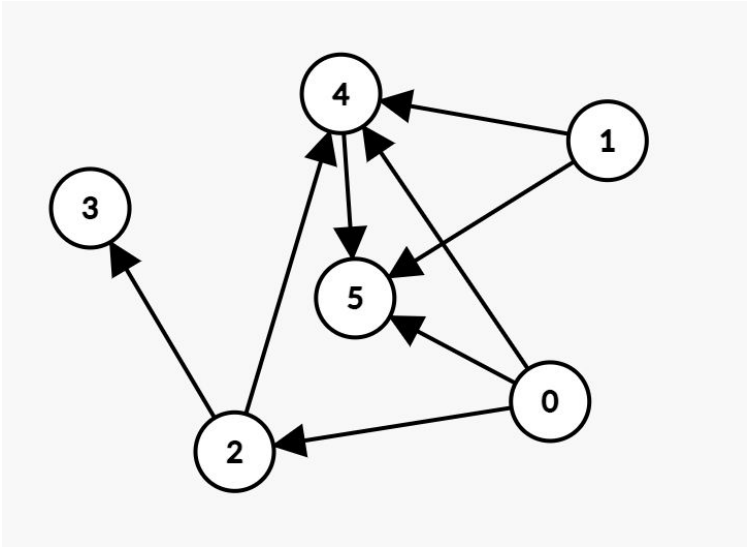
Unweighted Graph



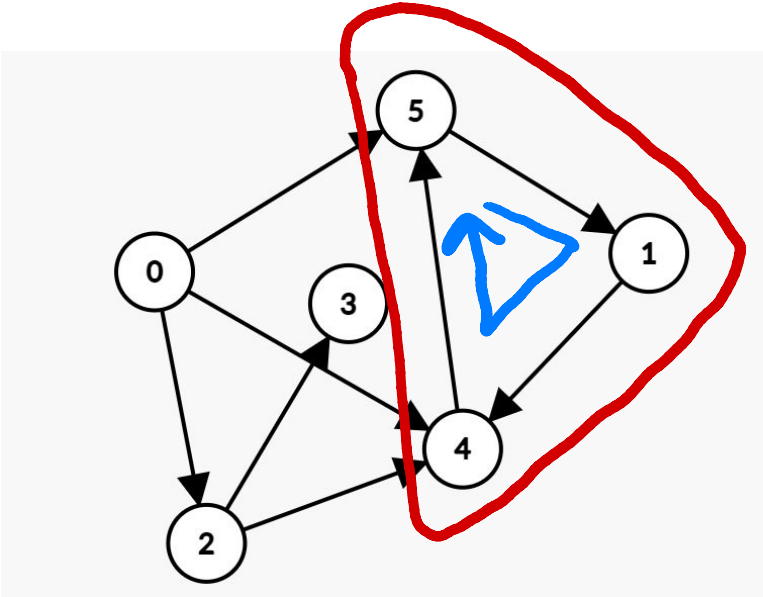
Weighted Graph



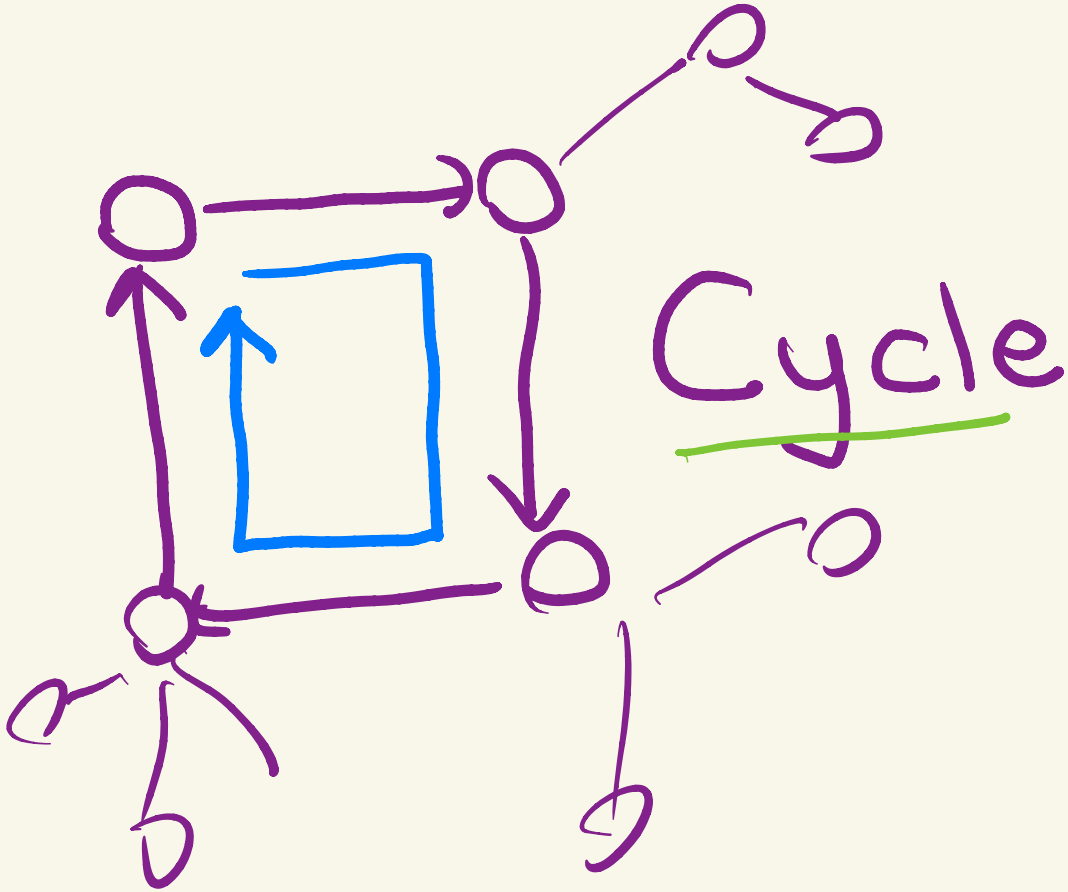
Cyclic and Acyclic Graph



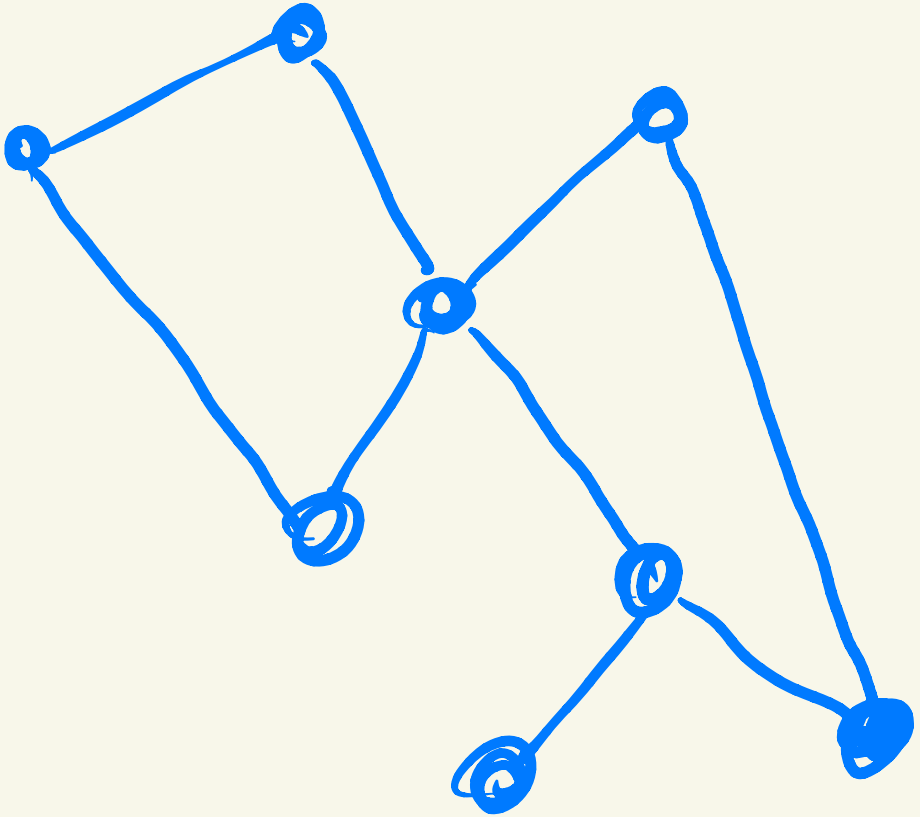
Acyclic Graph



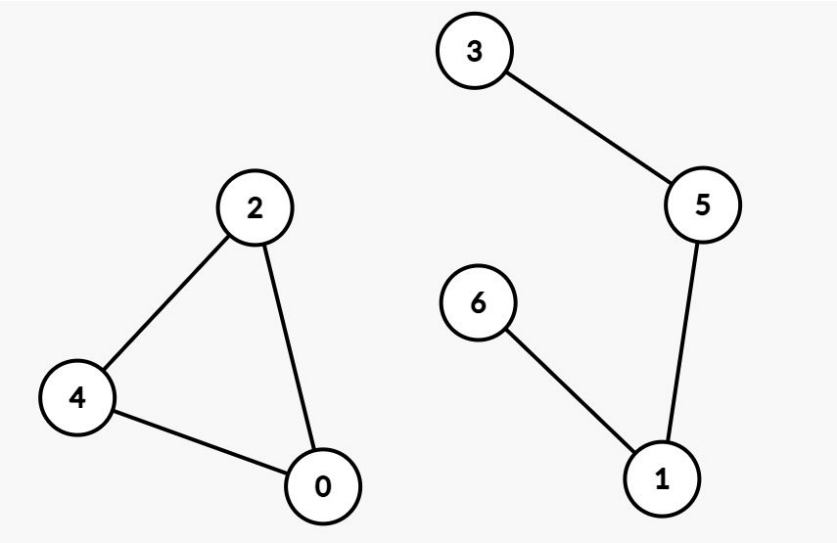
Cyclic Graph



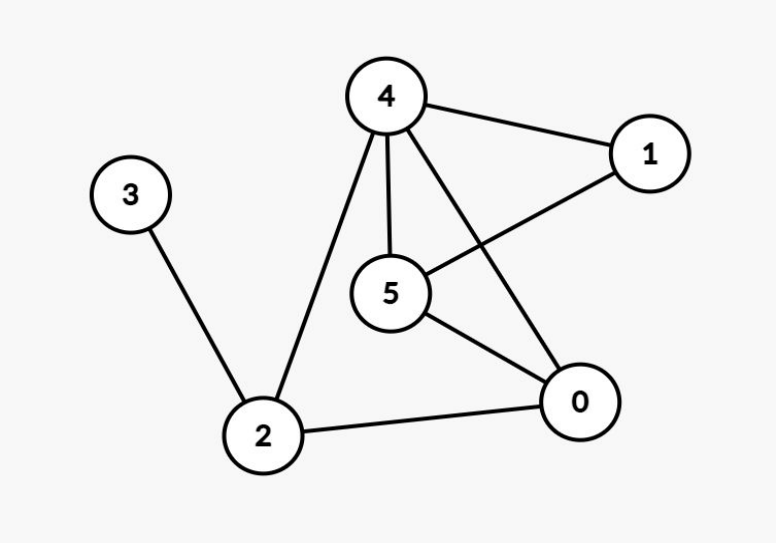
Cycle in undirected graph



Connected and Disconnected Graph

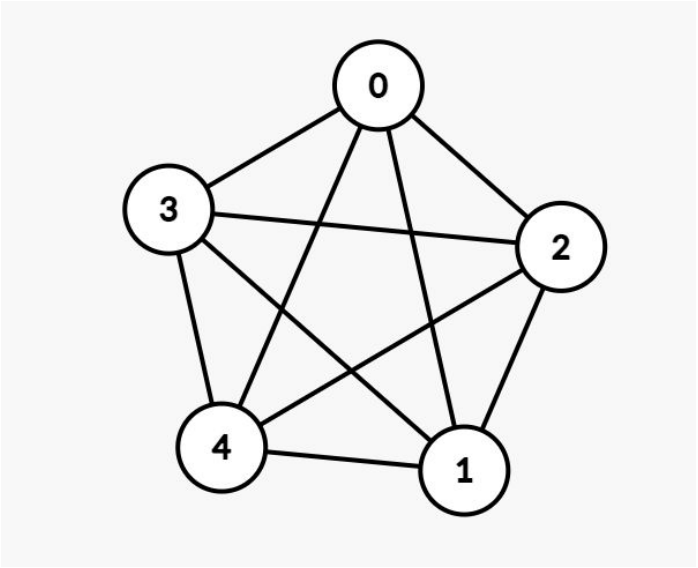


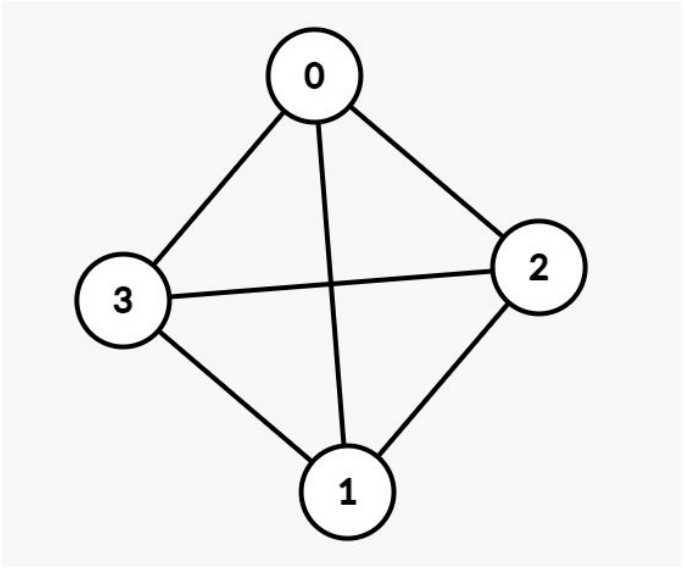
Disconnected Graph



Connected Graph

Complete Graph

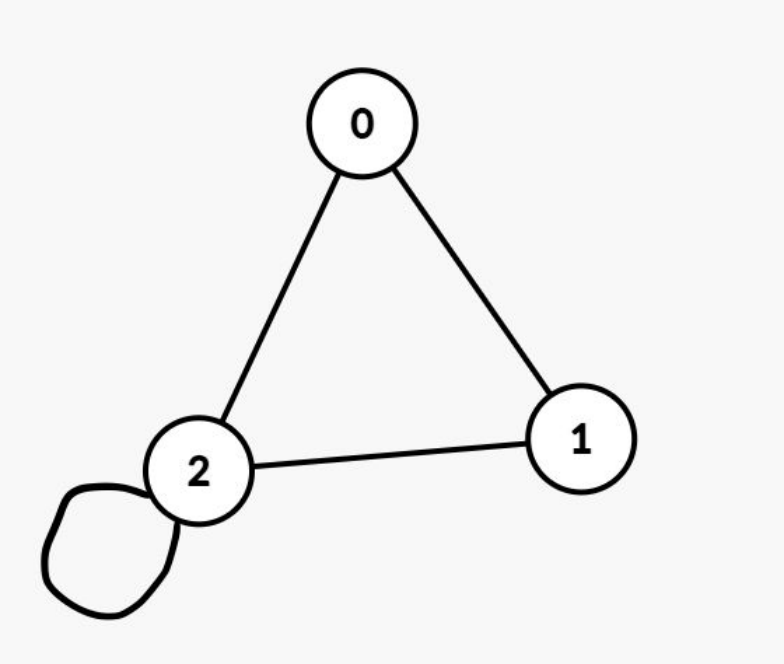




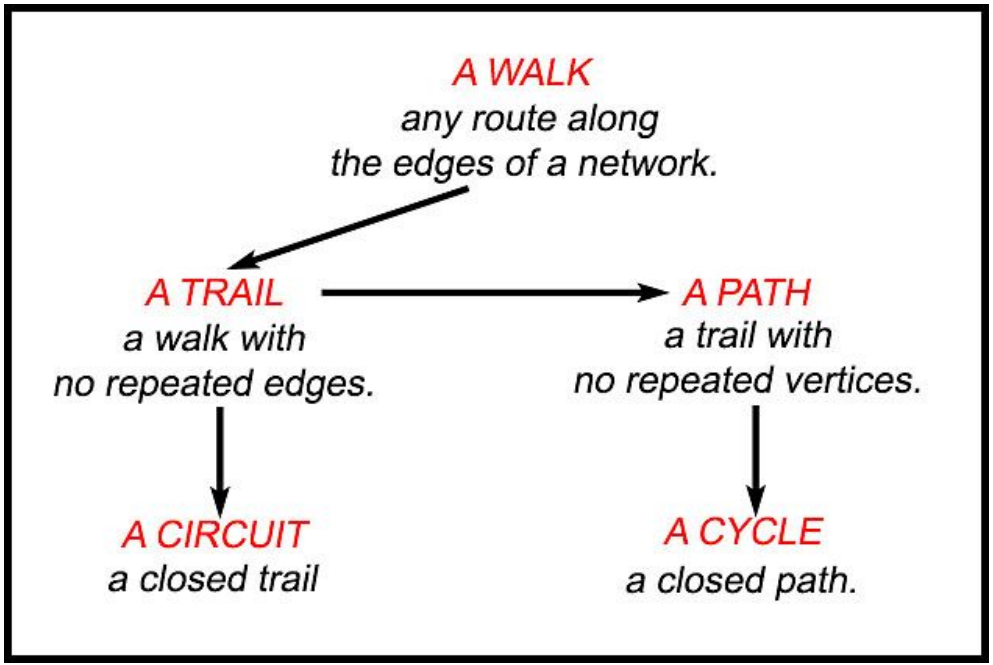
Common Terms

- Neighbours and Degree
- Self Loop
- Path and Walk
- Cycle
- Simple Path
- Articulation Points and Bridges

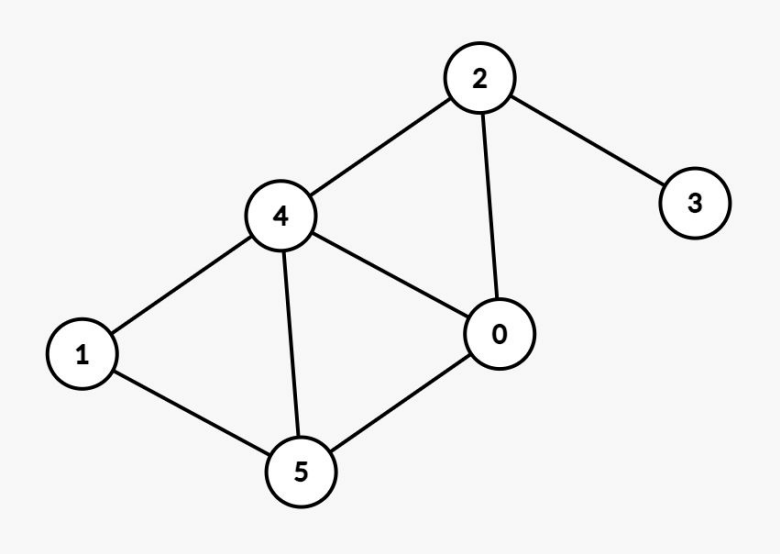
Self Loop



Path, Walk, Cycle, Circuit and More

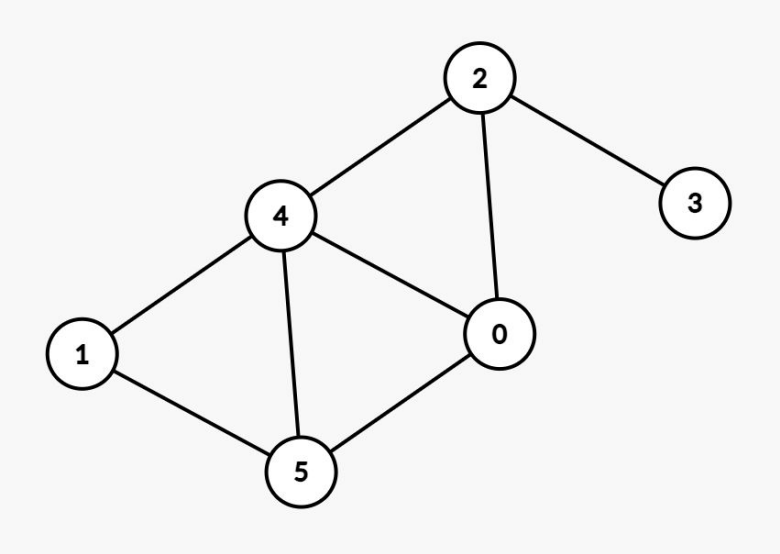


Example of Walk



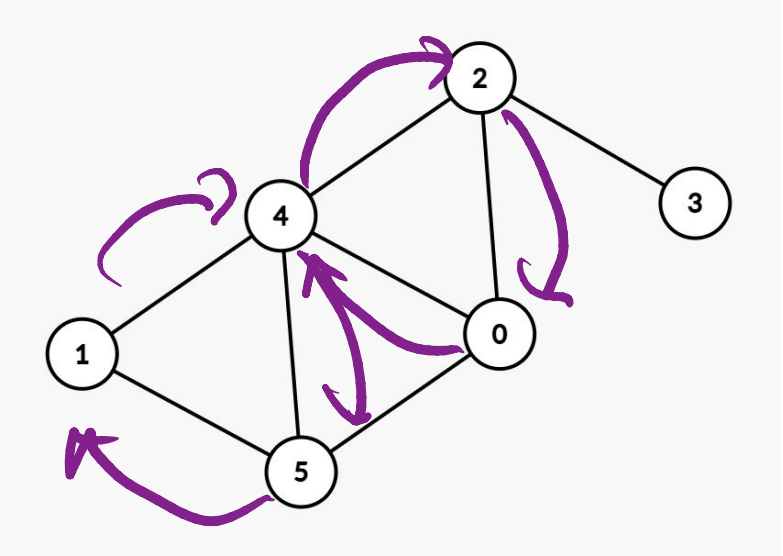
0 -> 4 -> 0 -> 2 -> 4 -> 5

Example of Trail



0 -> 4 -> 5 -> 0 -> 2 -> 4

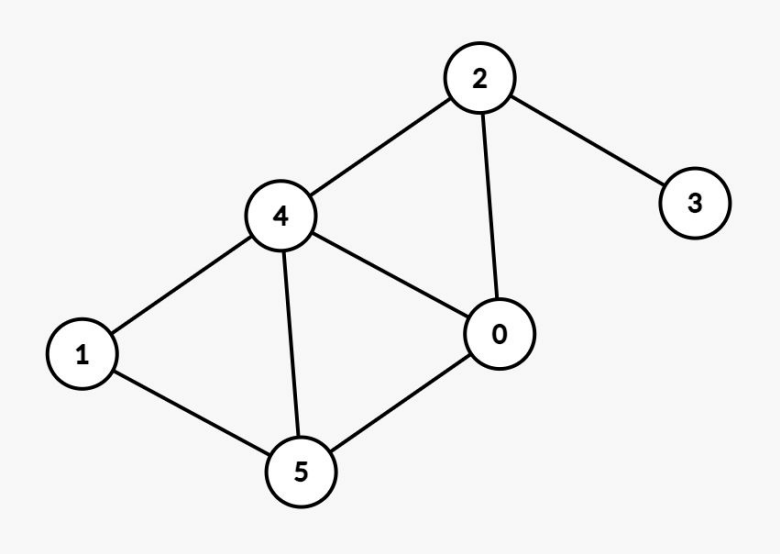
Example of Circuit



4 → 2 → 0 → 4
→ 5 → 1 → 4

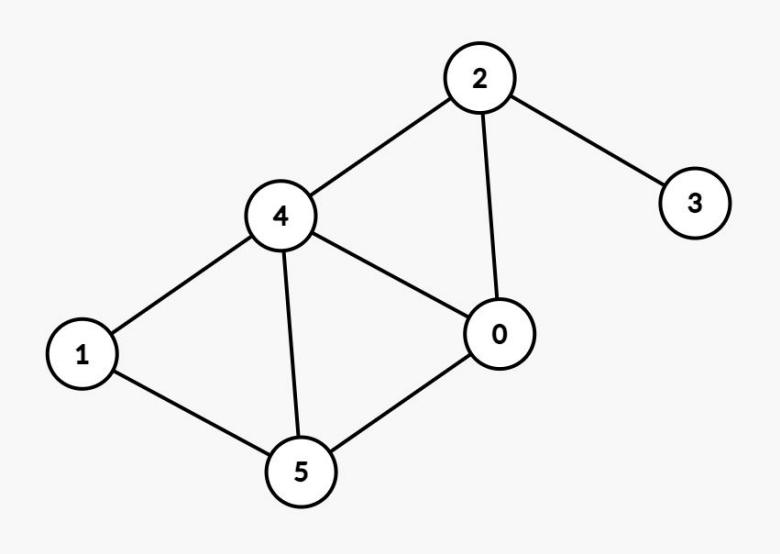
~~0 -> 2 -> 4 -> 0 -> 5 -> 0~~

Example of Path



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

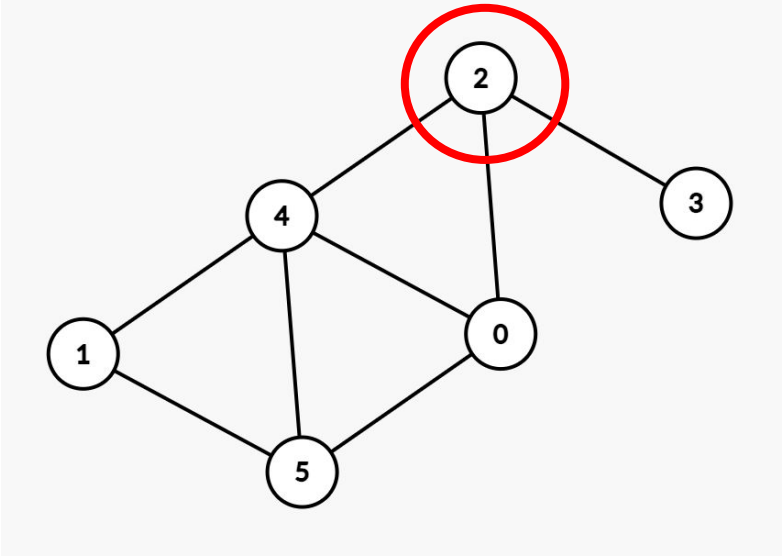
Example of Cycle



2 -> 0 -> 5 -> 1 -> 4 -> 2

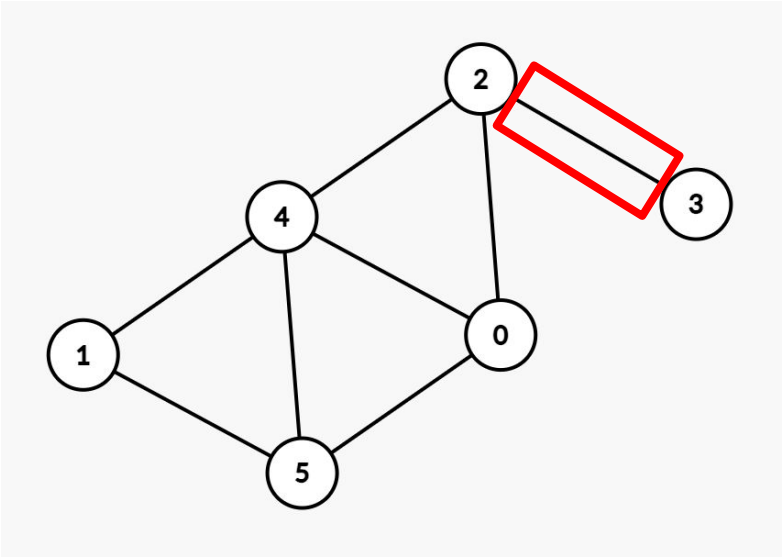
Articulation Point

If removing a node from the graph results in increase of connected components, then that node is called an articulation point.



Bridges

An edge in a graph when removed increases the number of connected components in the graph is said to be a bridge.



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
- The sum of all degrees in an undirected graph is even

Proof:

max edges in acyclic, connected graph:

n - 1

→ You add 1 more = n edges

= 1 cycle.

now,

$$\sum \deg_i \geq 2 \cdot n$$

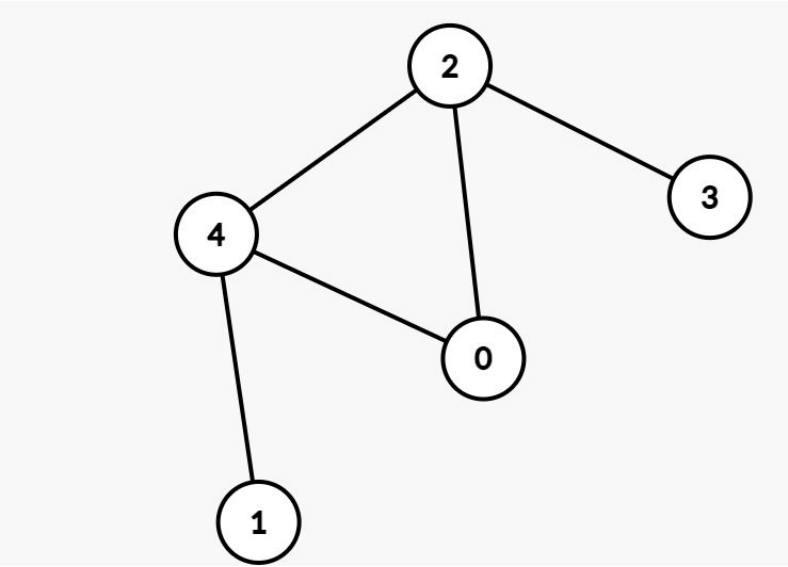
$$\therefore \text{edges} \geq \left\lfloor \frac{2n}{2} \right\rfloor \geq n$$

$\therefore$ if $\deg_i \geq 2$ for all i, cycle exists

Representation of Graphs

- Adjacency Matrix
- Adjacency List with Vector
- Adjacency List with Set

Adjacency Matrix



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

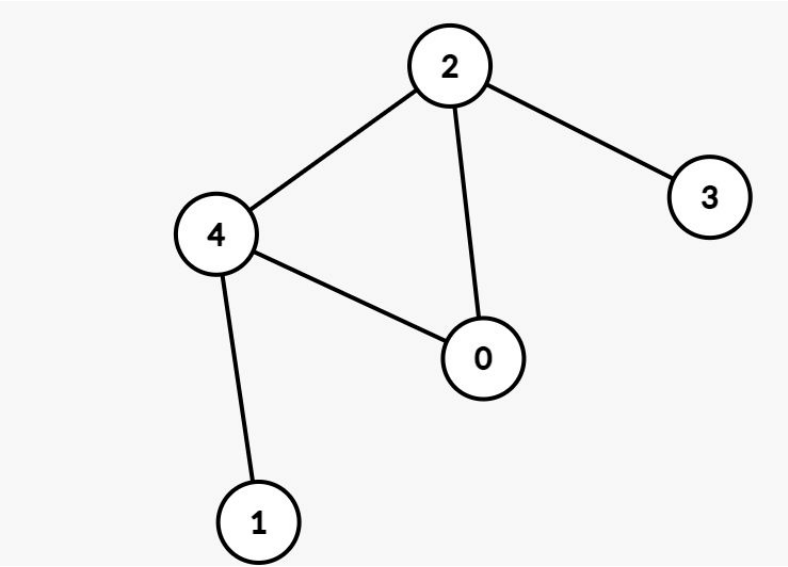
Implementation of Adjacency Matrix

```
int n, m; cin >> n >> m;

vector<vector<int>>adj(n, vector<int>(n));

for(int i = 0; i < m; i++) {
    int u, v; cin >> u >> v;
    adj[u][v] = adj[v][u] = 1;
}
```

Adjacency List



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

Implementation of Adjacency List (Vector)

```
int n, m; cin >> n >> m;

vector<vector<int>>adj(n);

for(int i = 0; i < m; i++) {
    int u, v; cin >> u >> v;
    adj[u].push_back(v);
    adj[v].push_back(u);
}
```

Implementation of Adjacency List (Set)

```
int n, m; cin >> n >> m;

vector<set<int>>adj(n);

for(int i = 0; i < m; i++) {
    int u, v; cin >> u >> v;
    adj[u].insert(v);
    adj[v].insert(u);
}
```

Problem to Consider

You are given a graph with N vertices and M edges. Perform Q queries on it.

Queries are of type:

1 i j : Add edge i to j in the graph

2 i j : Remove edge i to j in the Graph

3 i j : Print if an edge from i to j exists

4 i : Print number of neighbours of the node i

Trade-Off Analysis

	Adjacency Matrix	Adj List w/ Vector	Adj List w/ Set
Space Complexity	$O(n^2)$	O(edges)	O(edges)
Add an Edge	O(1)	O(1)	O(logn)
Remove an Edge	O(1)	O(n)	O(logn)
Search an Edge	O(1)	O(n)	O(logn)
No. of Neighbours	O(n)	O(1)	O(1)