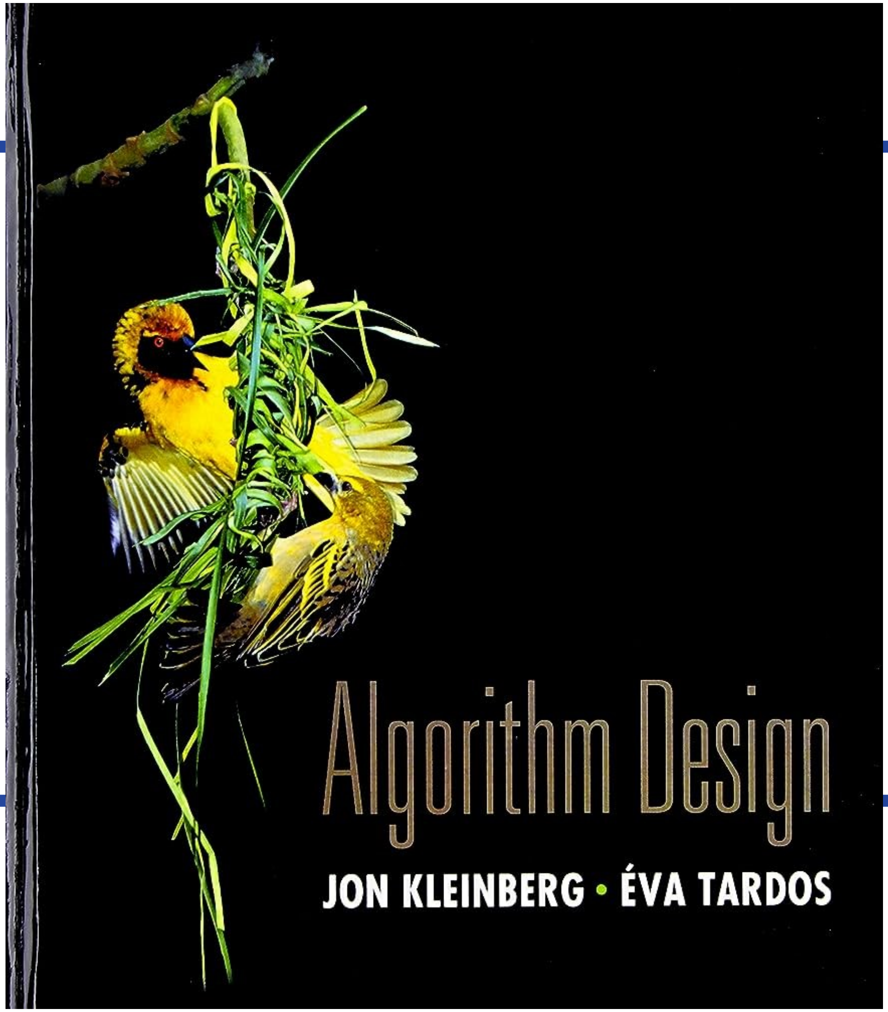
CS 310: Algorithms

Lecture 5

Instructor: Naveed Anwar Bhatti

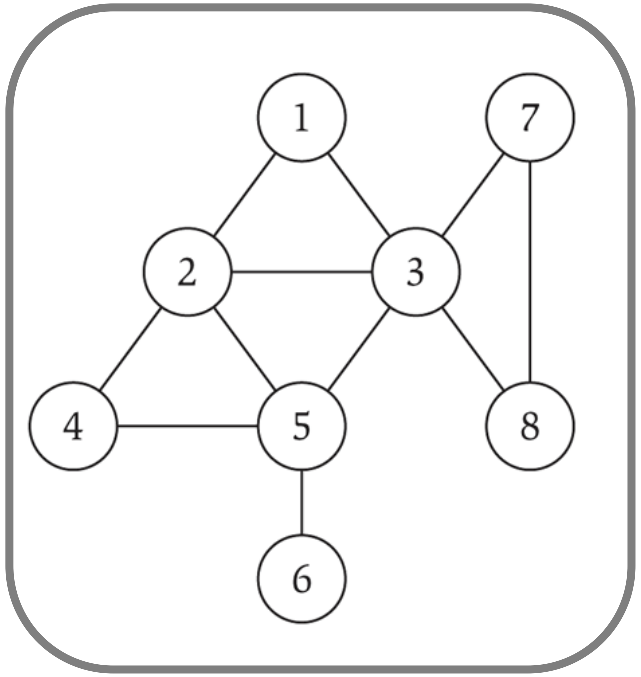


Chapter 3: Graphs

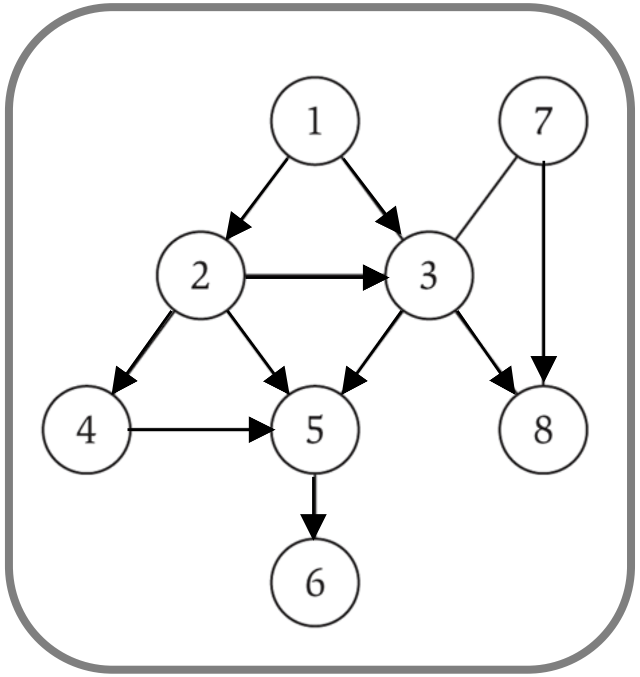
Section 3.1:
Basic Definitions and Applications

Graphs

Undirected

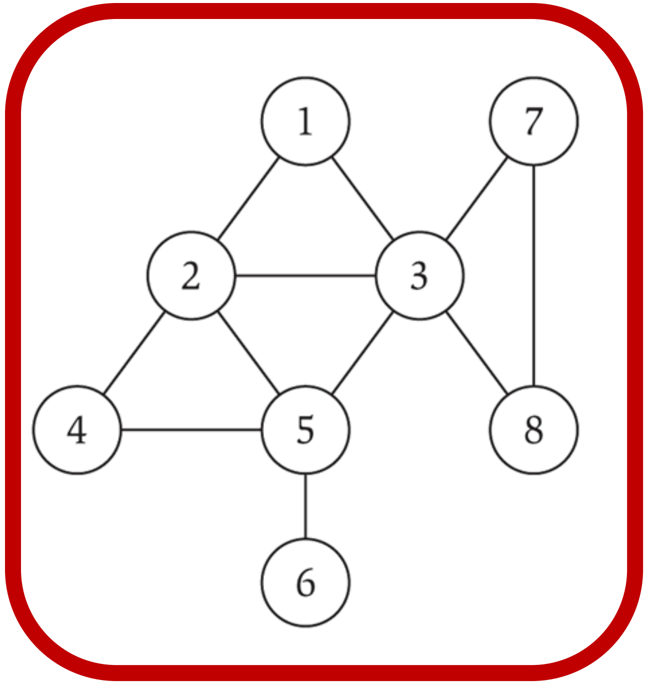


Directed

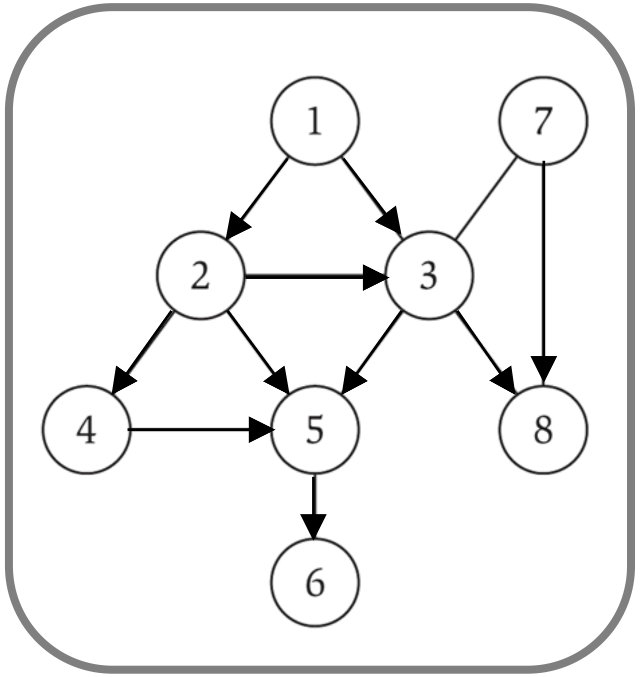


Graphs

Undirected

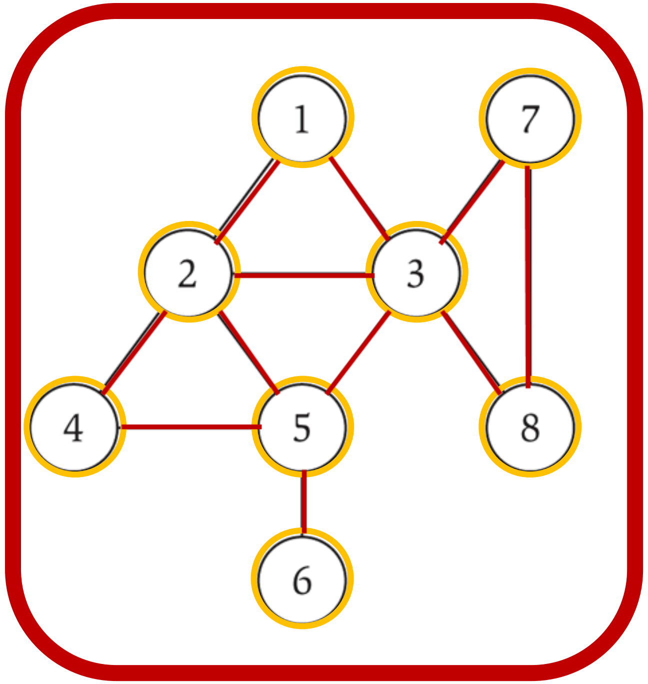


Directed



Graphs

Undirected



G = (V, E)

- V = nodes.
- E = edges between pairs of nodes.
- Captures **pairwise relationship** between objects.
- Graph size parameters: n = |V|, m = |E|.

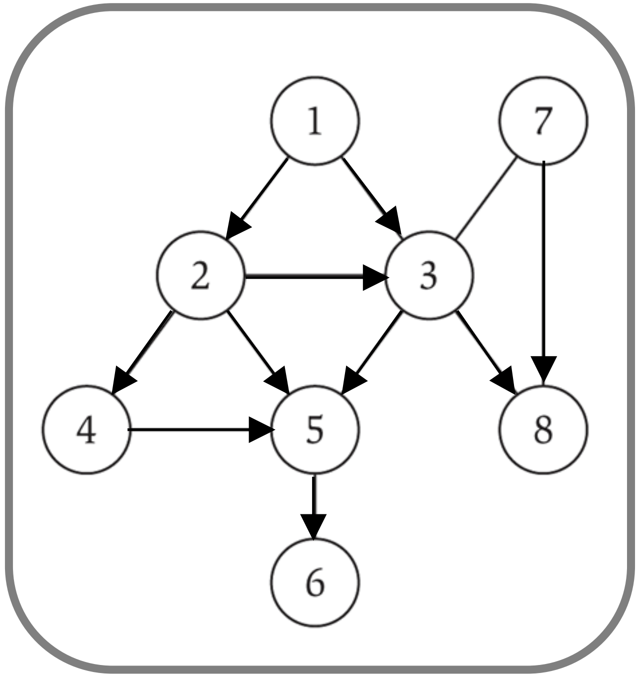
V = { 1, 2, 3, 4, 5, 6, 7, 8 }

E = { 1-2, 1-3, 2-3, 2-4, 2-5, 3-5, 3-7, 3-8, 4-5, 5-6, 7-8 }

n = 8 m = 11

Graphs

Directed



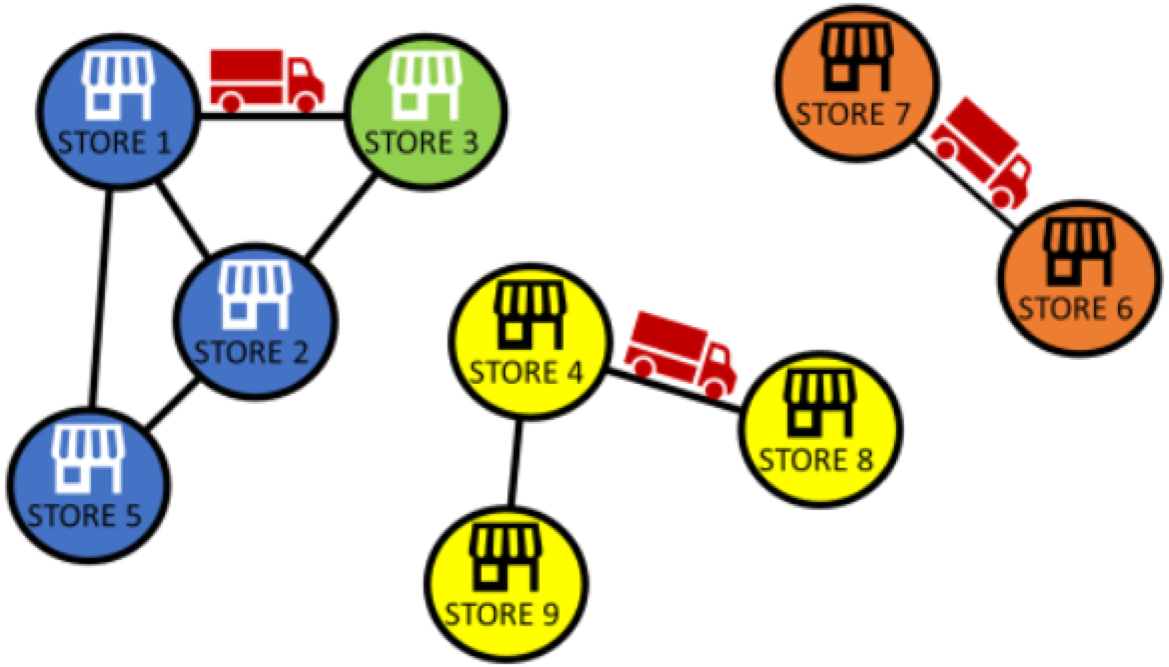
G = (V, E)

- V = nodes.
- E = edges between pairs of nodes.
- Captures **one-way relationship** between objects.
- Graph size parameters: n = |V|, m = |E|.

V = { 1, 2, 3, 4, 5, 6, 7, 8 }
E = { 1-2, 1-3, 2-3, 2-4, 2-5, 3-5, 3-7, 3-8, 4-5, 5-6, 7-8 }
n = 8
m = 11

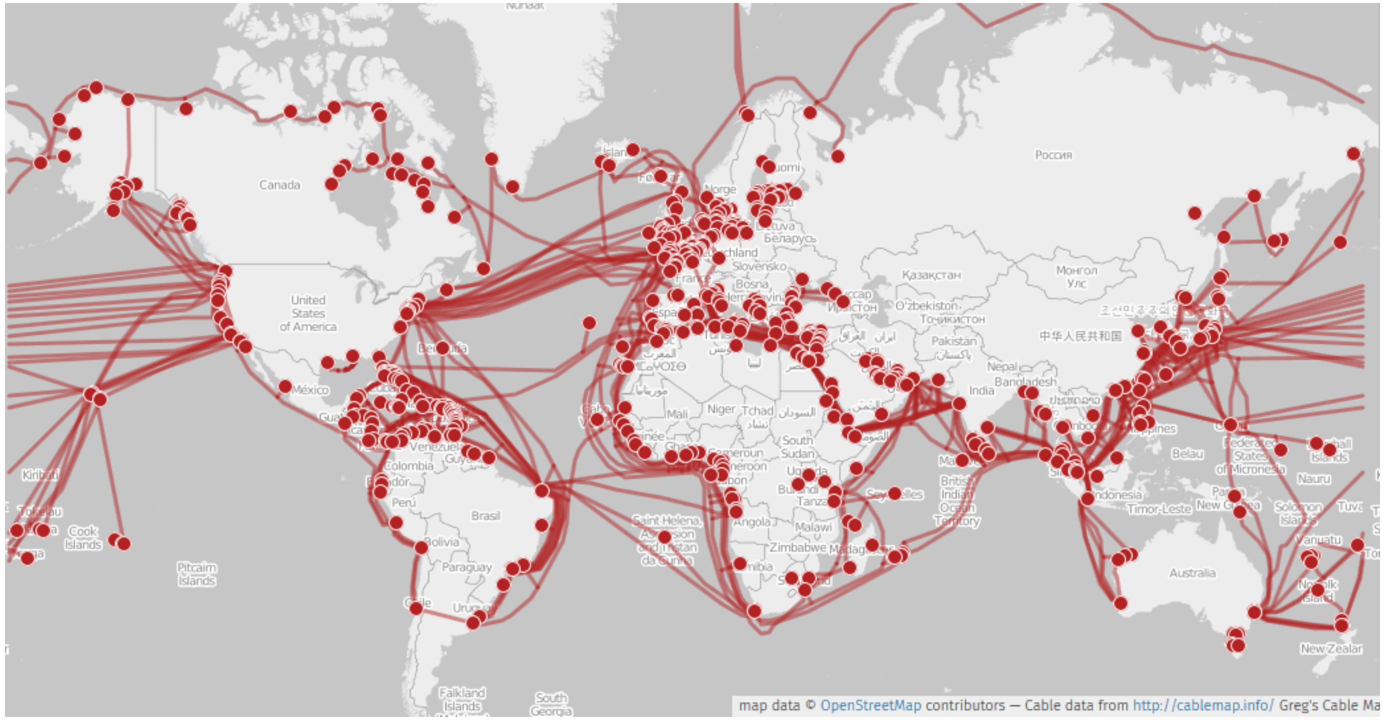
Some Graph Applications

Graph	*Nodes*	*Edges*
Transportation	street intersections	highways



Some Graph Applications

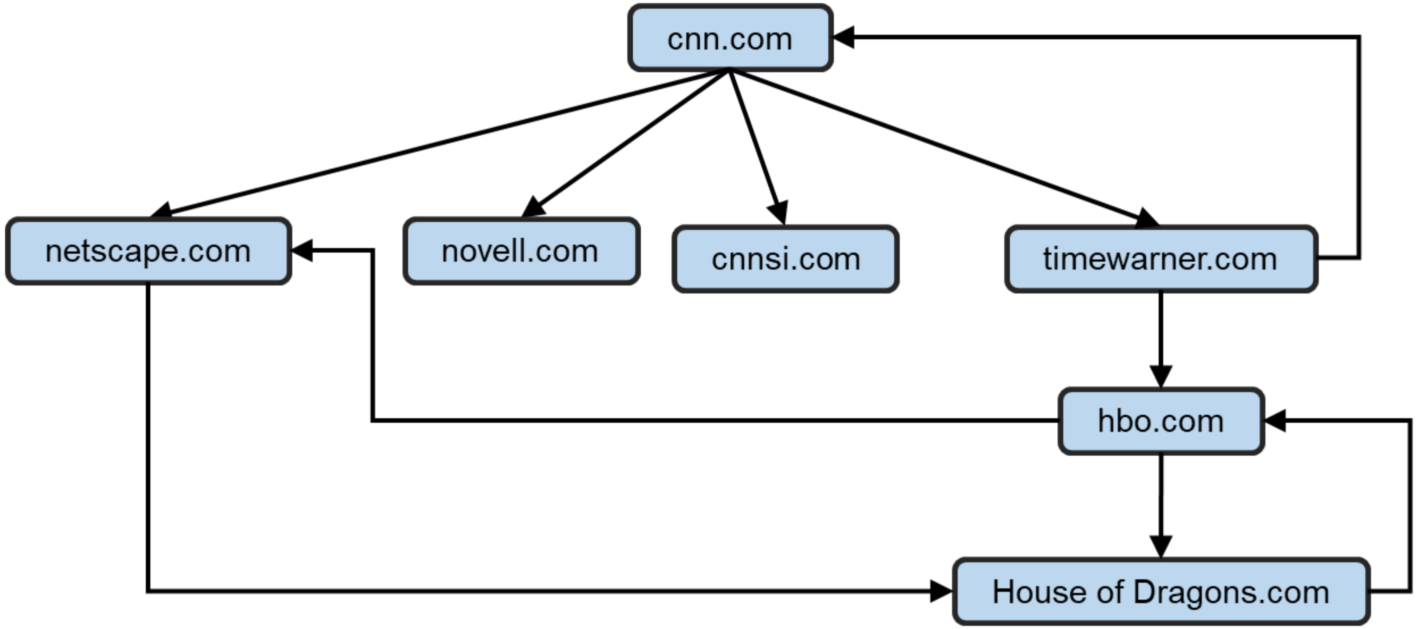
Graph	*Nodes*	*Edges*
Transportation	street intersections	highways
Communication	computers	fiber optic cables



map data © OpenStreetMap contributors — Cable data from http://cablemap.info/ Greg's Cable Ma

Some Graph Applications

Graph	*Nodes*	*Edges*
Transportation	street intersections	highways
Communication	computers	fiber optic cables
World Wide Web	web pages	hyperlinks

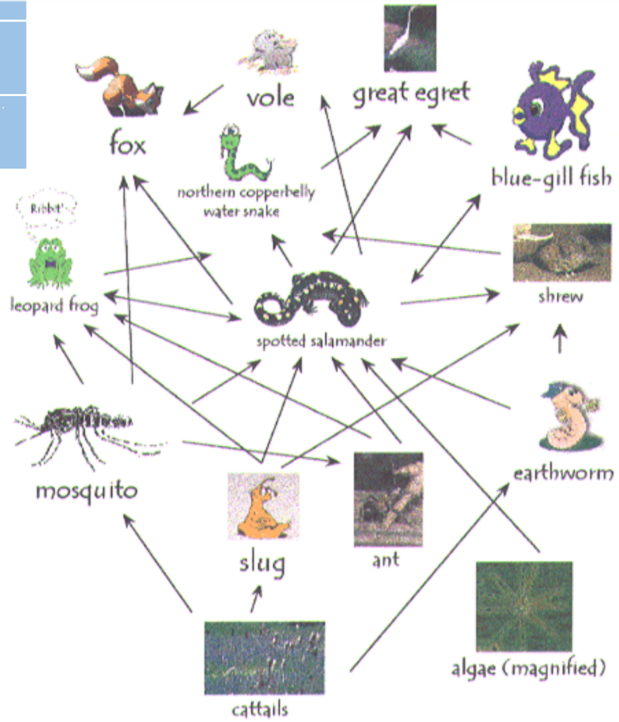


Some Graph Applications

Graph	*Nodes*	*Edges*
Transportation	street intersections	highways
Communication	computers	fiber optic cables
World Wide Web	web pages	hyperlinks
Social	people	relationships

Some Graph Applications

Graph	*Nodes*	*Edges*
Transportation	street intersections	highways
Communication	computers	fiber optic cables
World Wide Web	web pages	hyperlinks
Social	people	relationships
Food Web	species	predator-prey

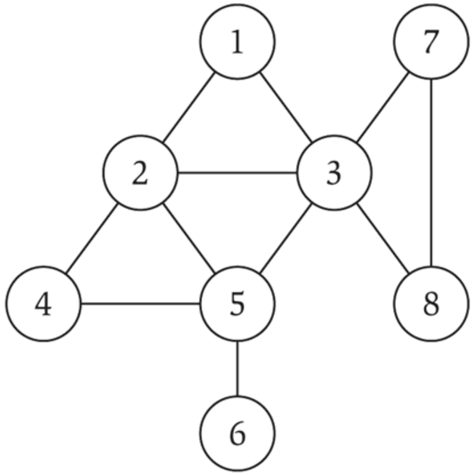


Some Graph Applications

Graph	*Nodes*	*Edges*
Transportation	street intersections	highways
Communication	computers	fiber optic cables
World Wide Web	web pages	hyperlinks
Social	people	relationships
Food Web	species	predator-prey
Software systems	functions	function calls
Scheduling	tasks	precedence constraints
Circuits	gates	wires

Graph Representation: Adjacency Matrix

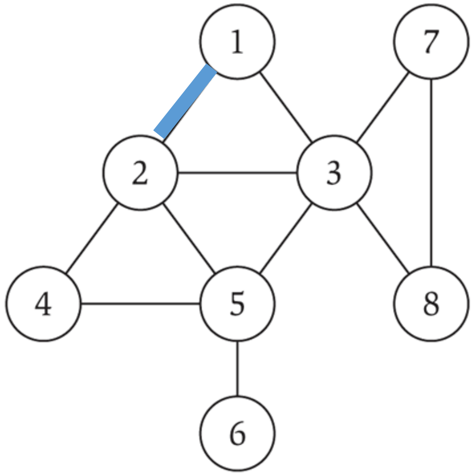
- Adjacency matrix. n-by-n matrix with $\mathbf{A_{uv} = 1}$ if (u, v) is an edge.
  - Two representations of each edge.
  - Space proportional to n^2.



	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graph Representation: Adjacency Matrix

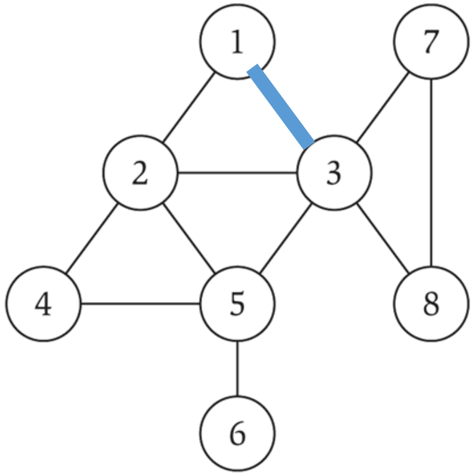
- Adjacency matrix. n-by-n matrix with $\mathbf{A_{uv} = 1}$ if (u, v) is an edge.
  - Two representations of each edge.
  - Space proportional to n^2.



	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graph Representation: Adjacency Matrix

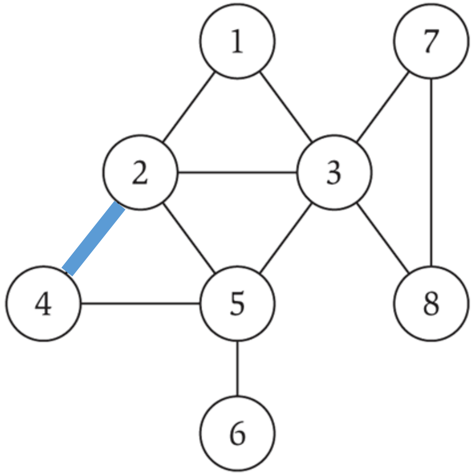
- Adjacency matrix. n-by-n matrix with $\mathbf{A_{uv} = 1}$ if (u, v) is an edge.
  - Two representations of each edge.
  - Space proportional to n^2.



	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graph Representation: Adjacency Matrix

- Adjacency matrix. n-by-n matrix with $\mathbf{A_{uv} = 1}$ if (u, v) is an edge.
  - Two representations of each edge.
  - Space proportional to n^2.



	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graphs: Live Poll 1

Given an adjacency matrix for a graph of size N, which mathematical expression represents the number of iterations required to find all the edges for both directed and undirected graphs with no self-loops?
(I'm not talking about Big O)

A. Directed **n^2** , Undirected **n^2**

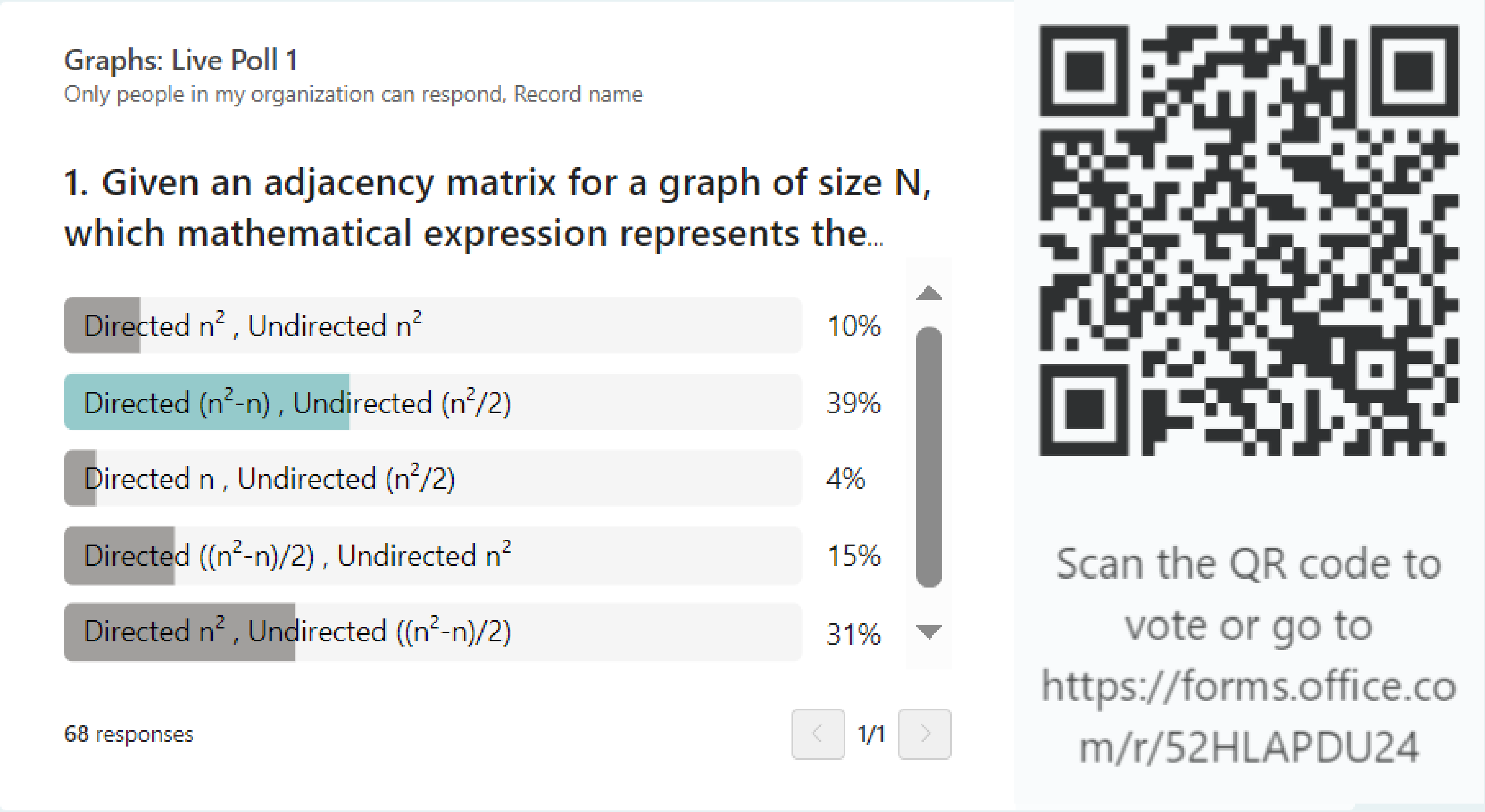
B. Directed **(n^2-n)** , Undirected **$(n^2/2)$**

C. Directed **n** , Undirected **$(n^2/2)$**

D. Directed **$((n^2-n)/2)$** , Undirected **n^2**

E. Directed **n^2** , Undirected **$((n^2-n)/2)$**

Scan the QR code to vote or go to https://forms.office.com/r/52HLAPDU24



Graphs: Live Poll 1

Only people in my organization can respond, Record name

1. Given an adjacency matrix for a graph of size N, which mathematical expression represents the...

Directed n^2 , Undirected n^2	10%
Directed (n^2-n) , Undirected $(n^2/2)$	39%
Directed n , Undirected $(n^2/2)$	4%
Directed $((n^2-n)/2)$, Undirected n^2	15%
Directed n^2 , Undirected $((n^2-n)/2)$	31%

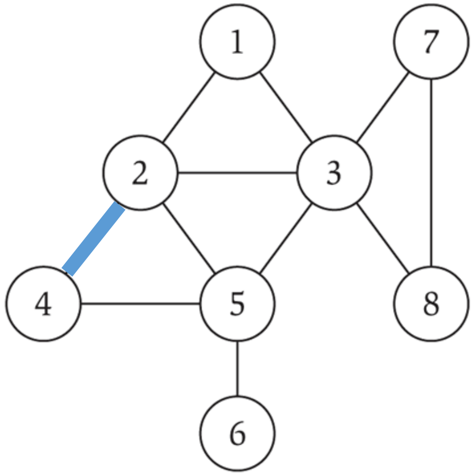
68 responses

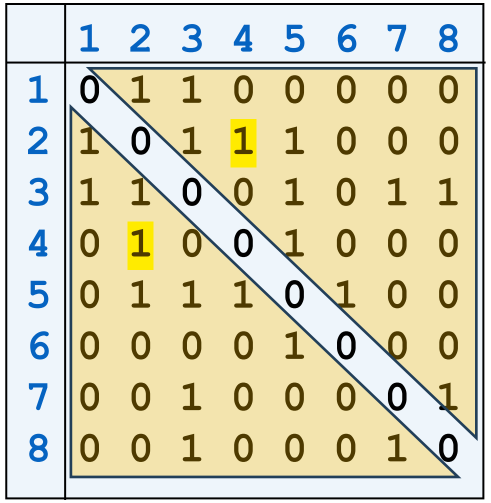
1/1

Scan the QR code to vote or go to https://forms.office.com/r/52HLAPDU24

Graph Representation: Adjacency Matrix

- Adjacency matrix. n-by-n matrix with A_{uv} = 1 if (u, v) is an edge.
  - Two representations of each edge.
  - Space proportional to n^2.

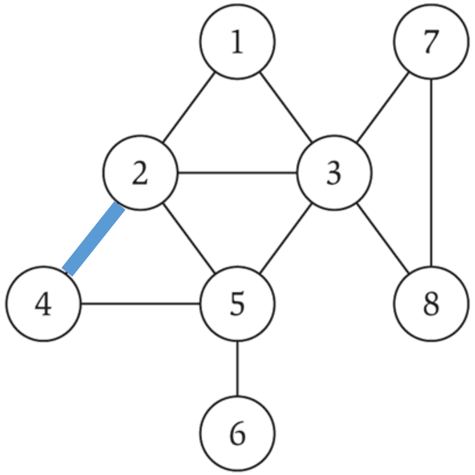




	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graph Representation: Adjacency Matrix

- Adjacency matrix. n-by-n matrix with A_{uv} = 1 if (u, v) is an edge.
  - Two representations of each edge.
  - Space proportional to n^2.
  - Checking if (u, v) is an edge takes $\Theta(1)$ time.
  - Identifying all edges takes $\Theta(n^2)$ time.



	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	1	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graphs: Live Poll 2

Given an undirected graph represented by an adjacency matrix, if the sum of all elements in the matrix is 'S', and the graph has no self-loops, how many edges does the graph have?

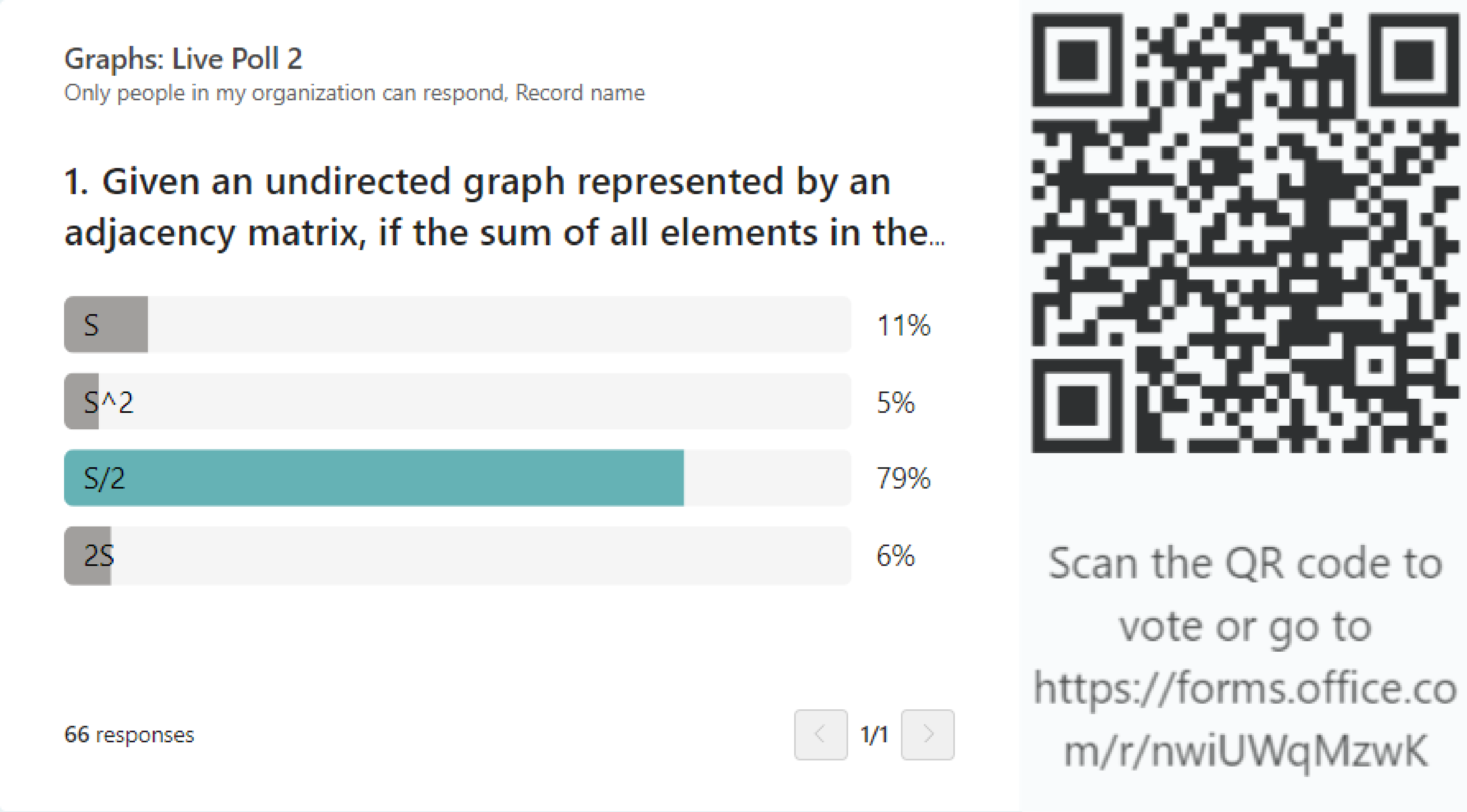
A. S

$B.\ S^2$

C. S/2

D. 2S

Scan the QR code to vote or go to https://forms.office.com/r/nwiUWqMzwK



Graphs: Live Poll 2

Only people in my organization can respond, Record name

1. Given an undirected graph represented by an adjacency matrix, if the sum of all elements in the...

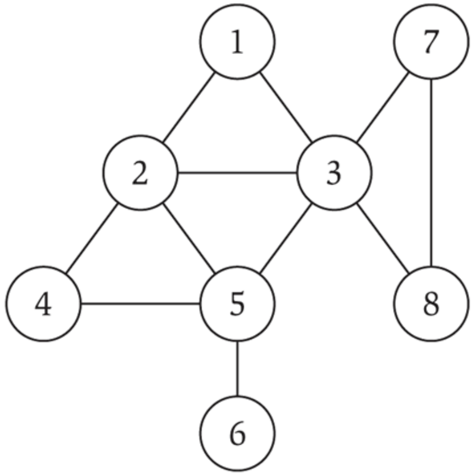
S	11%
S^2	5%
S/2	79%
2S	6%

66 responses

Scan the QR code to vote or go to https://forms.office.com/r/nwiUWqMzwK

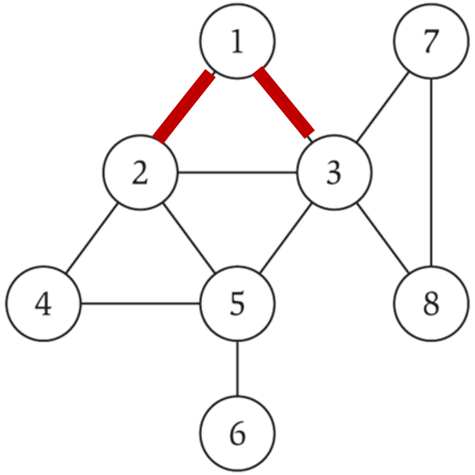
Graph Representation: Adjacency List

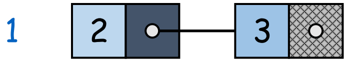
- Adjacency list.



Graph Representation: Adjacency List

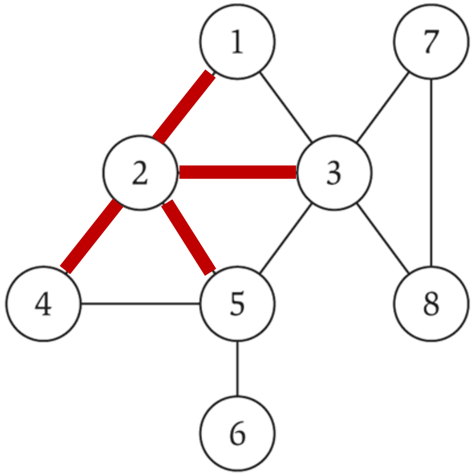
- Adjacency list.

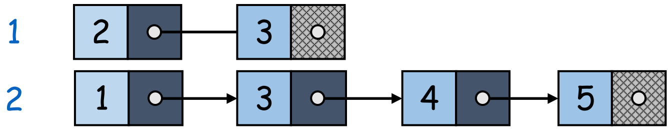




Graph Representation: Adjacency List

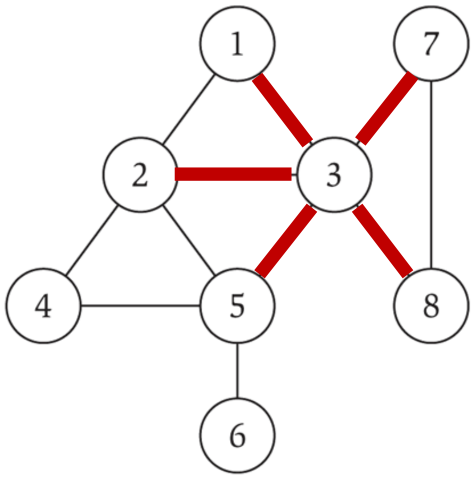
- Adjacency list.

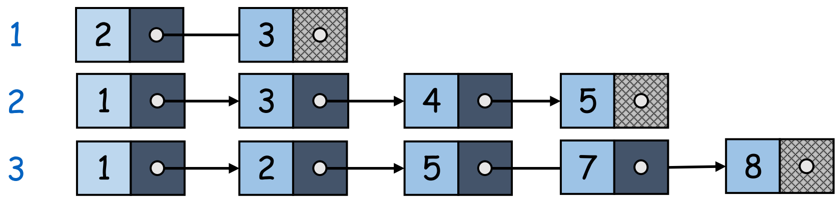



1 → 2 → 3
2 → 1 → 3 → 4 → 5

Graph Representation: Adjacency List

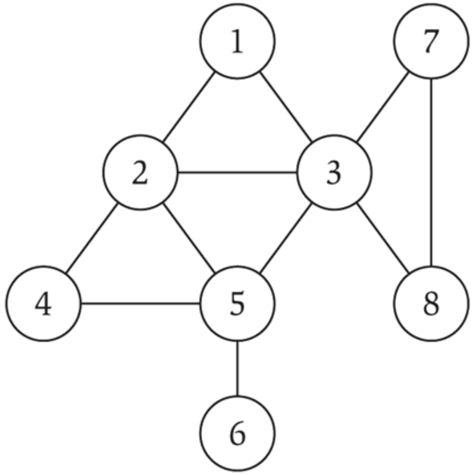
- Adjacency list.

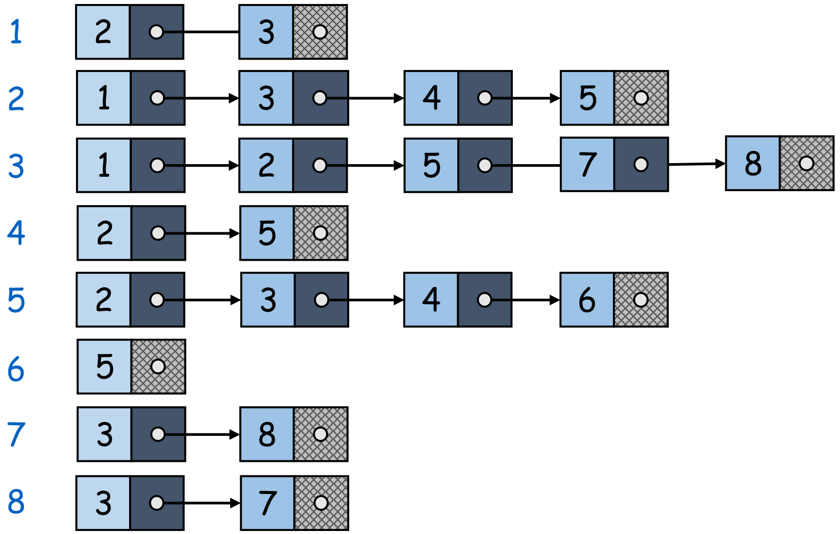




Graph Representation: Adjacency List

- Adjacency list.





1 2 → 3

2 1 → 3 → 4 → 5

3 1 → 2 → 5 → 7 → 8

4 2 → 5

5 2 → 3 → 4 → 6

6 5

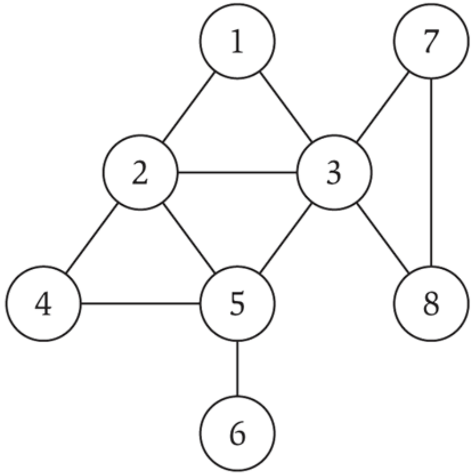
7 3 → 8

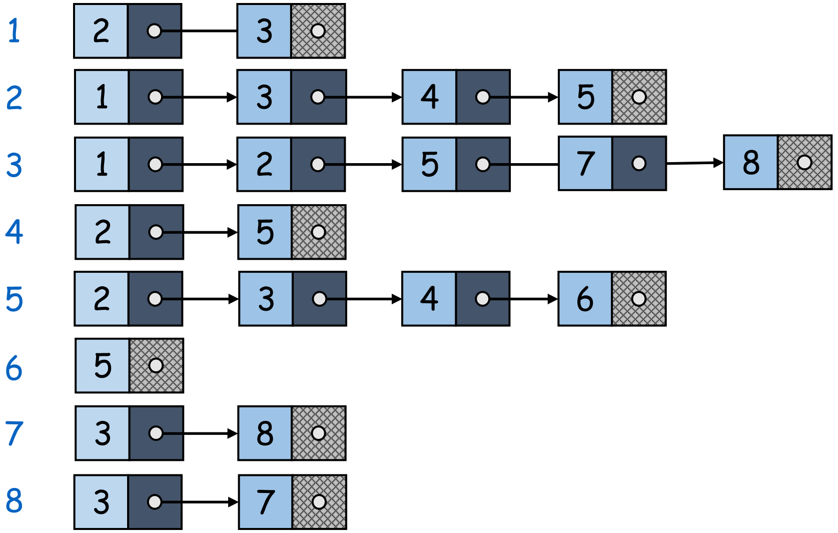
8 3 → 7

Graph Representation: Adjacency List

- Adjacency list.
  - Two representations of each edge.
  - Space proportional to **2m + n**.
  - Checking if (u, v) is an edge takes **O(deg(u))** time.
  - Identifying all edges takes Θ(2m + n)= Θ(m + n) time.

degree = number of neighbors of u





1: 2 → 3

2: 1 → 3 → 4 → 5

3: 1 → 2 → 5 → 7 → 8

4: 2 → 5

5: 2 → 3 → 4 → 6

6: 5

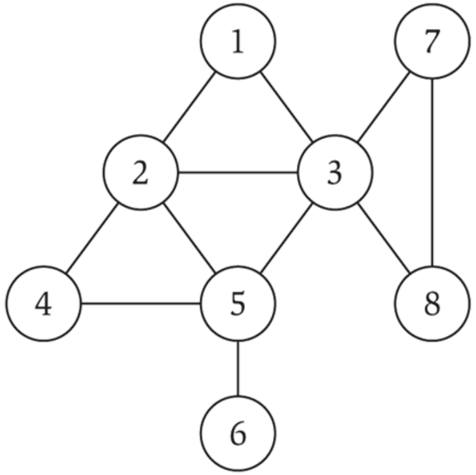
7: 3 → 8

8: 3 → 7

Paths and Connectivity

- **Def:** A path in an undirected graph G = (V, E) is a sequence ***P*** of nodes v_1, v_2, …, v_{k-1}, v_k with the property that each consecutive pair v_i, v_{i+1} is joined by an edge in E.

- **Def:** A path is simple if all nodes are distinct.

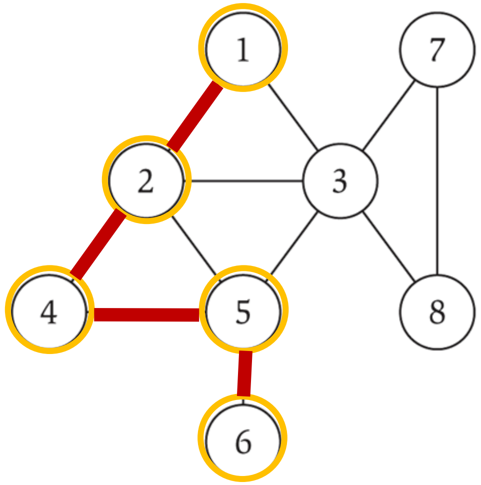


Paths and Connectivity

- **Def:** A path in an undirected graph G = (V, E) is a sequence P of nodes v_1, v_2, …, v_{k-1}, v_k with the property that each consecutive pair v_i, v_{i+1} is joined by an edge in E.

- **Def:** A path is simple if all nodes are distinct.

Path A = 1-2-4-5-6



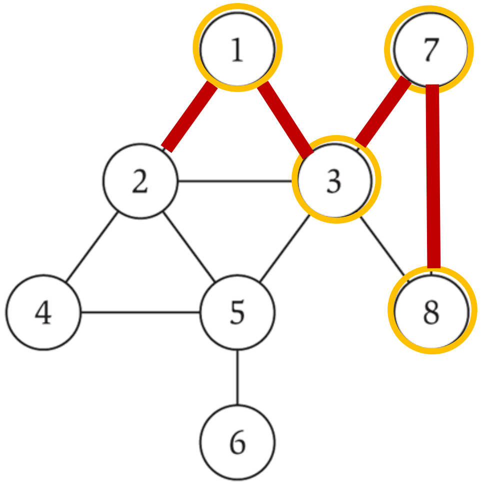
Paths and Connectivity

- **Def:** A path in an undirected graph G = (V, E) is a sequence P of nodes v_1, v_2, …, v_{k-1}, v_k with the property that each consecutive pair v_i, v_{i+1} is joined by an edge in E.

- **Def:** A path is simple if all nodes are distinct.

Path A = 1-2-4-5-6

Path B = 1-3-7-8

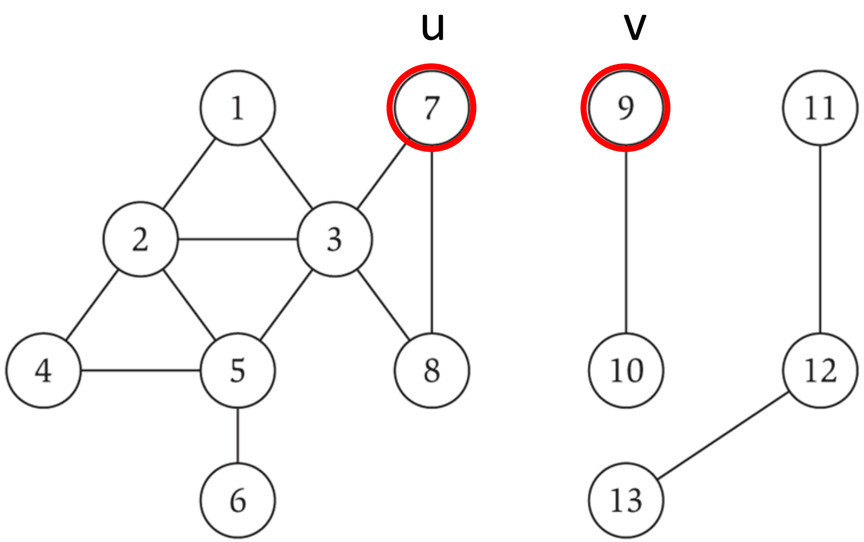


Paths and Connectivity

- **Def:** A path in an undirected graph G = (V, E) is a sequence P of nodes v_1, v_2, …, v_{k-1}, v_k with the property that each consecutive pair v_i, v_{i+1} is joined by an edge in E.

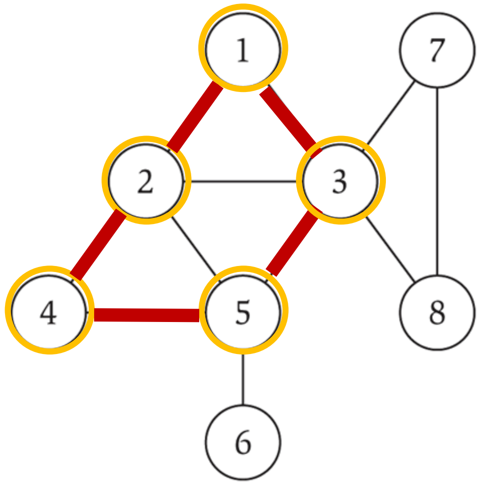
- **Def:** A path is simple if all nodes are distinct.

- **Def:** An undirected graph is connected if for every pair of nodes u and v, there is a path between u and v.



Cycles

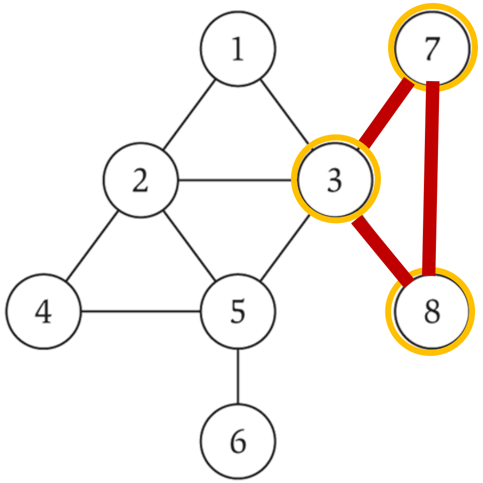
- **Def:** A cycle is a path $v_1, v_2, \ldots, v_{k-1}, v_k$ in which $v_1 = v_k$, $k > 2$, and the first k-1 nodes are all distinct.



cycle A = 1-2-4-5-3-1

Cycles

- **Def:** A cycle is a path $v_1, v_2, \ldots, v_{k-1}, v_k$ in which $v_1 = v_k$, $k > 2$, and the first k-1 nodes are all distinct.

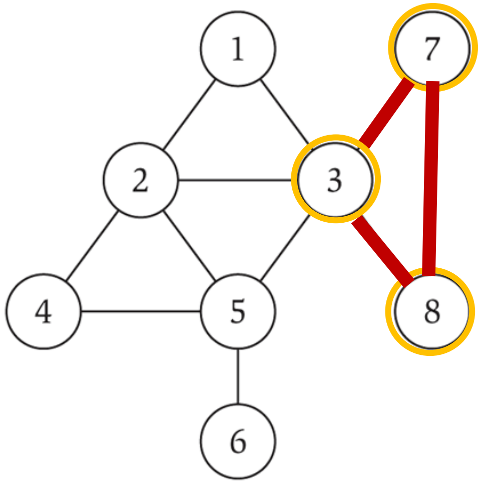


cycle A = 1-2-4-5-3-1

cycle B = 7-3-8-7

Cycles

- A graph with **n-1 edges** can not have **cycles**?

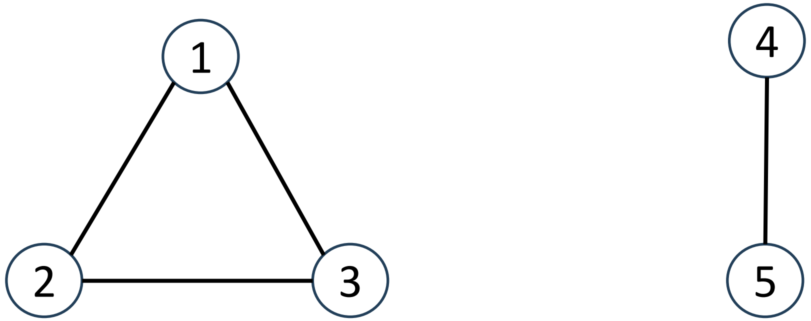


cycle A = 1-2-4-5-3-1

cycle B = 7-3-8-7

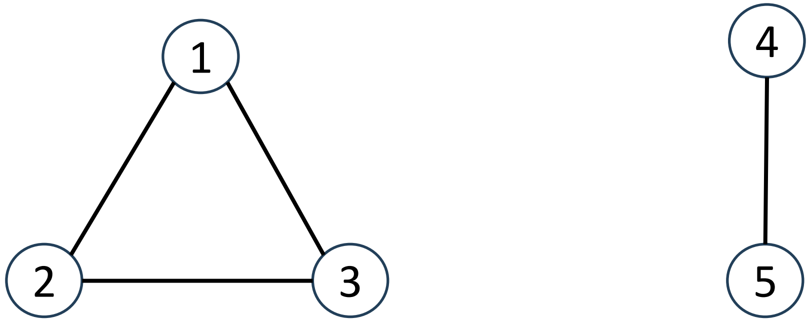
Cycles

- A graph with **n-1 edges** can not have **cycles**?



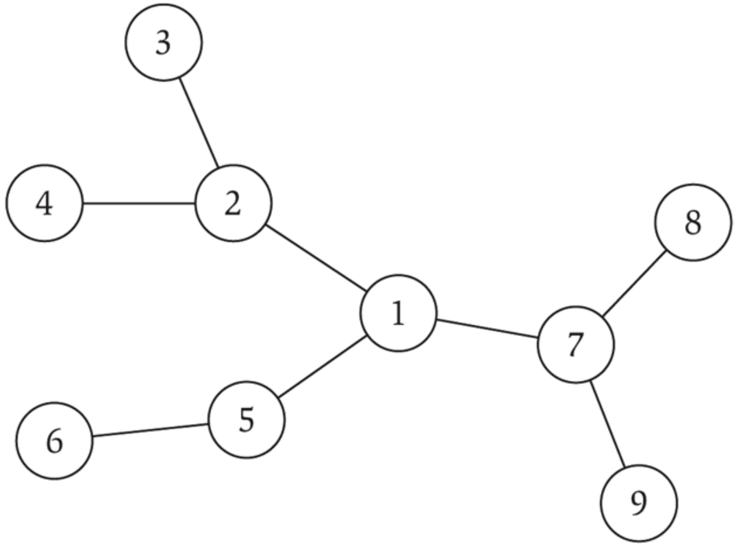
Cycles

- A **connected graph** with **n-1 edges** can not have **cycles**



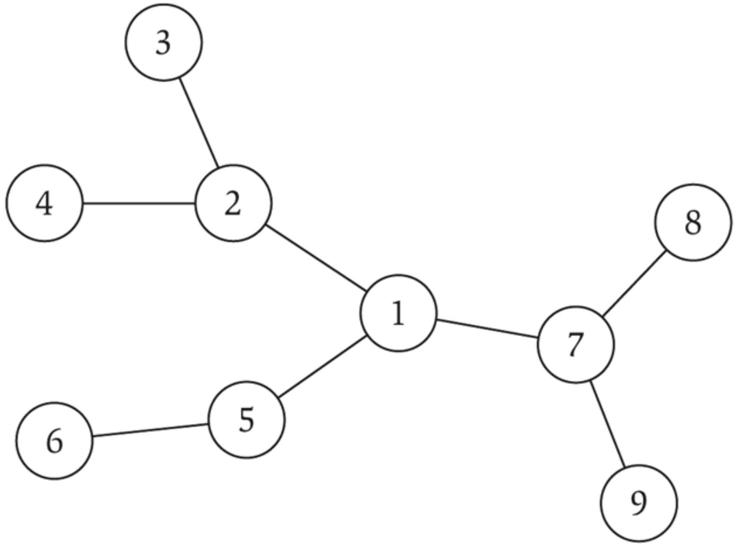
Graphs

- **Theorem:** Let **G** be an undirected graph on **n** nodes. Any two of the following statements imply the third.
  - G is connected.
  - G does not contain a cycle.
  - G has n-1 edges.



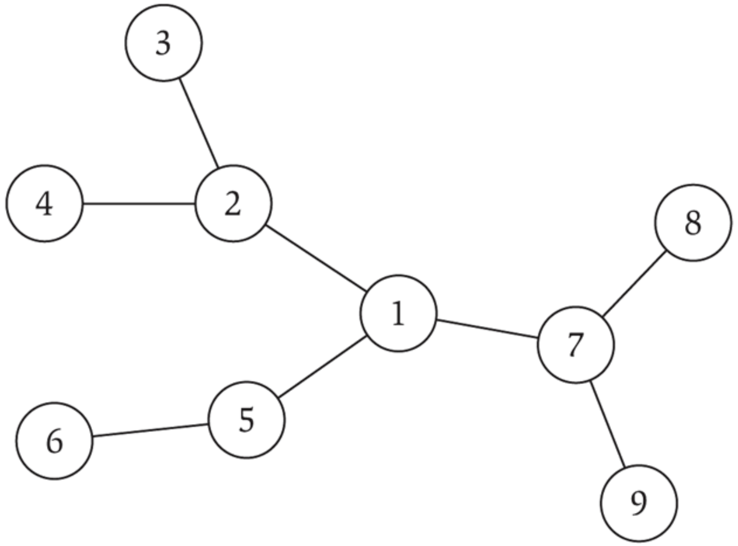
Trees

- **Theorem:** Let **G** be an undirected graph on **n** nodes. Any two of the following statements imply the third.
  - **G is connected.**
  - **G does not contain a cycle.**
  - G has n-1 edges.



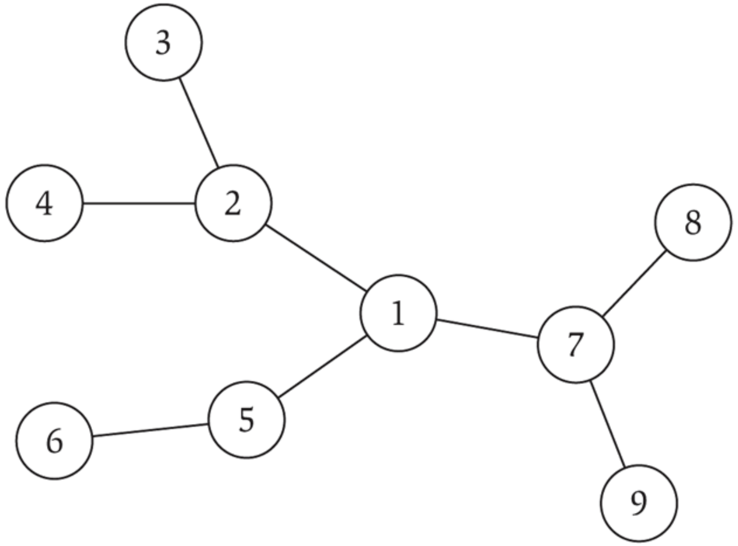
Trees

- **Theorem:** Let **G** be an undirected graph on **n** nodes. Any two of the following statements imply the third.
  - G is connected.
  - **G does not contain a cycle.**
  - **G has n-1 edges.**



Trees

- **Theorem:** Let **G** be an undirected graph on **n** nodes. Any two of the following statements imply the third.
  - **G is connected.**
  - G does not contain a cycle.
  - **G has n-1 edges.**



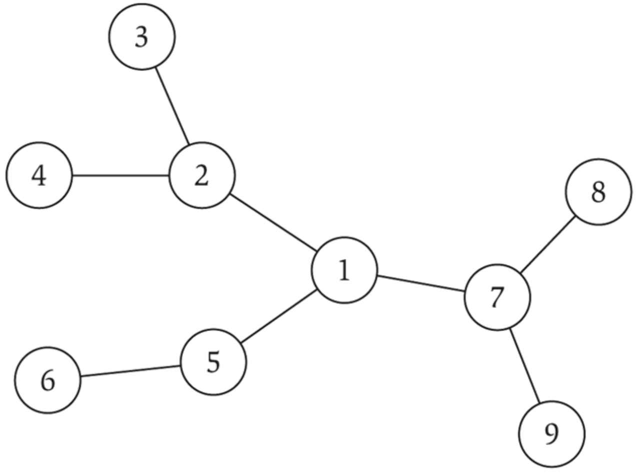
Trees

- **Def:** An undirected graph is a tree if it is **connected** and **does not contain a cycle**.

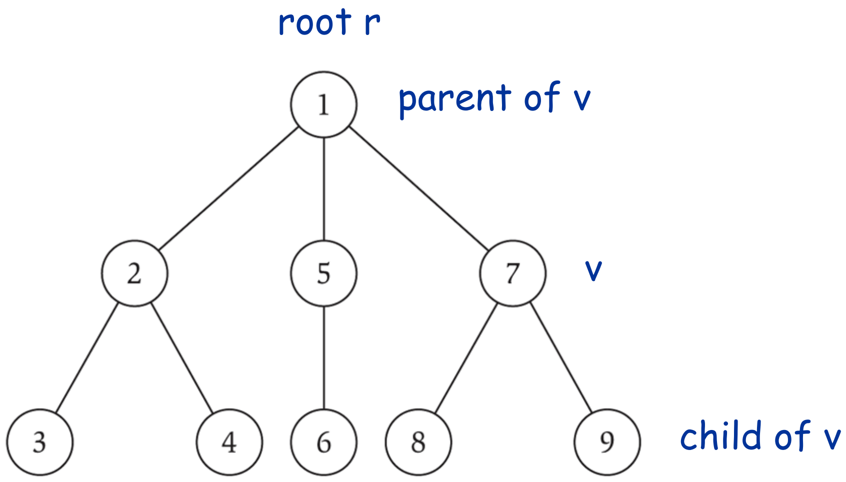
Trees

- **Def:** An undirected graph is a tree if it is **connected** and **does not contain a cycle**.

- **Rooted tree:** Given a tree **T**, choose a root node **r** and orient each edge away from r.



a tree

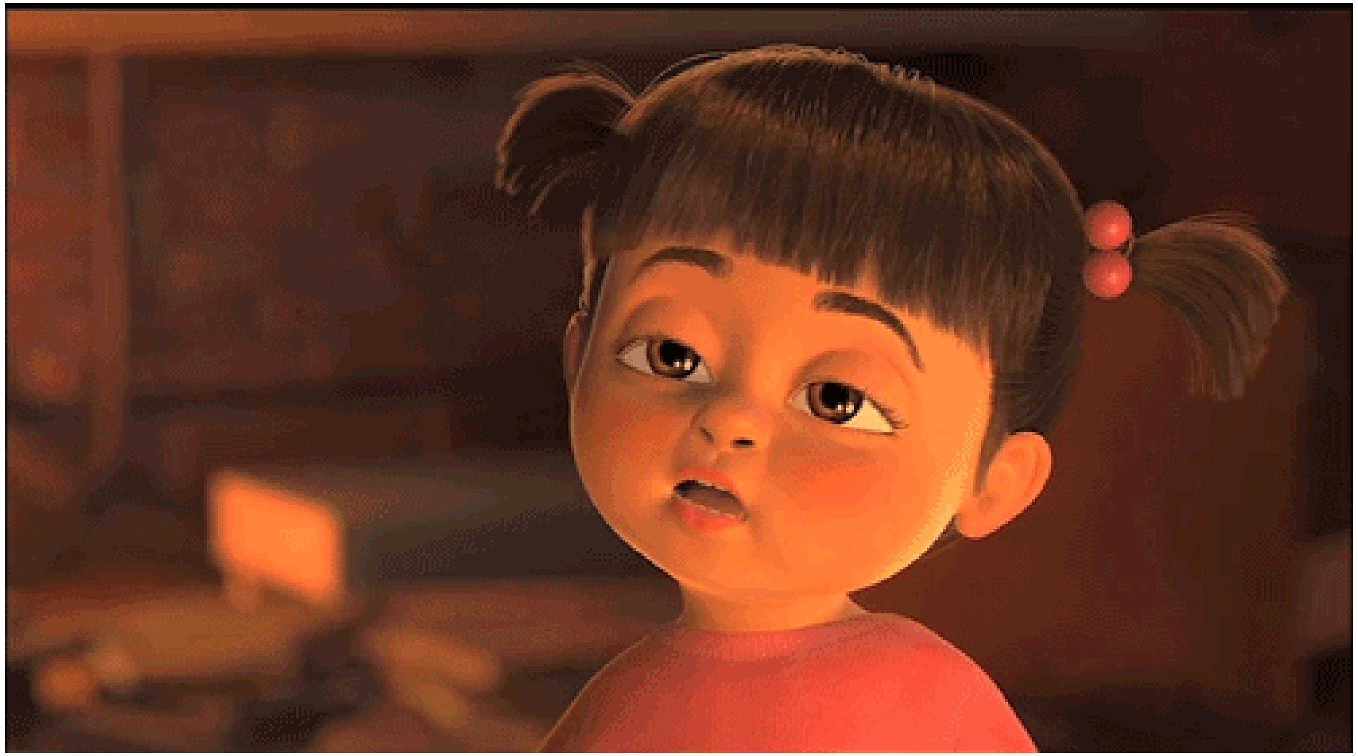


the same tree, rooted at 1

Section 3.2:
Graph Traversal

Next Time

Thanks a lot



If you are taking a Nap, **wake up**........Lecture Over