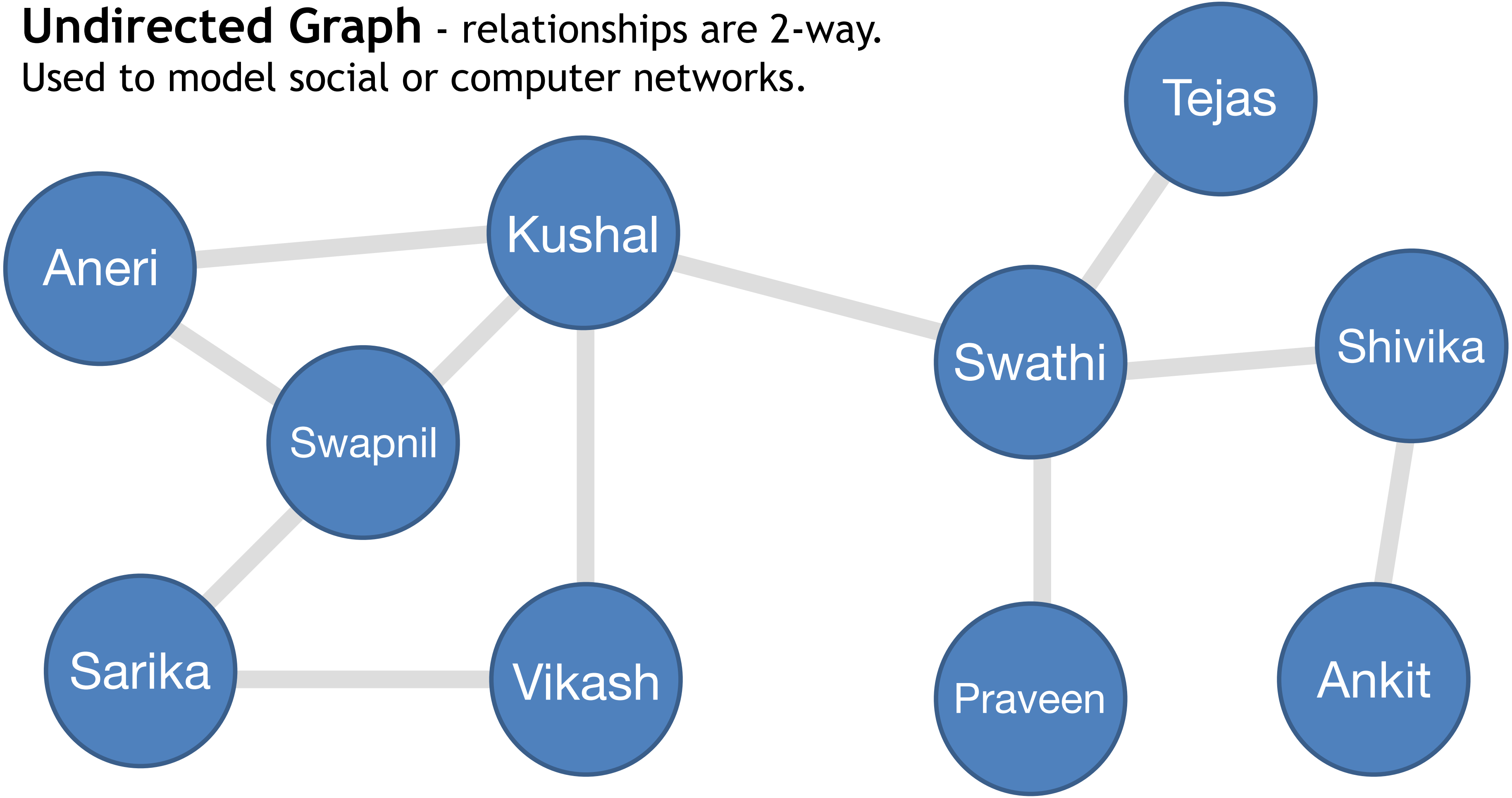
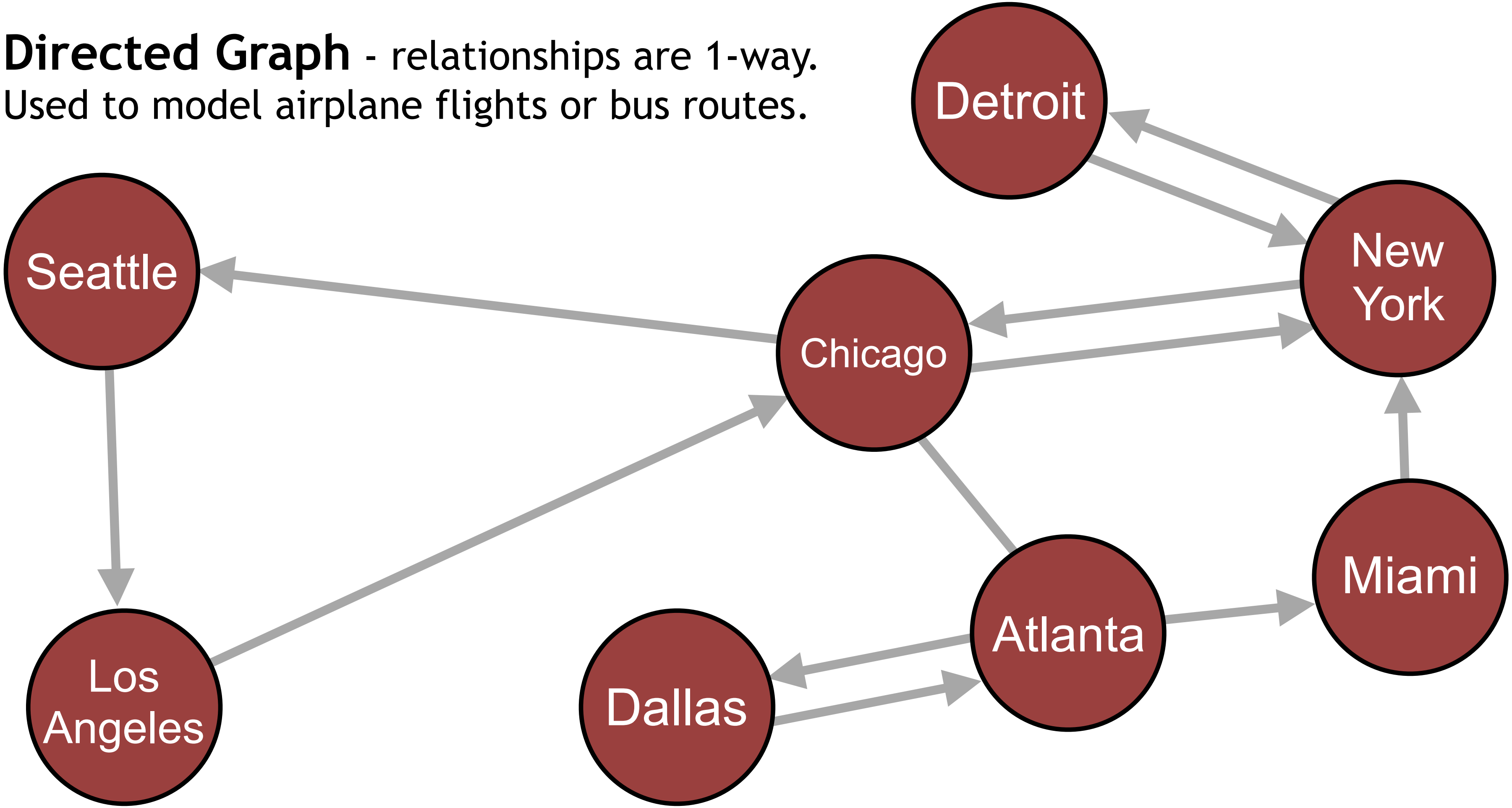
# Graphs

**Undirected Graph** - relationships are 2-way.
Used to model social or computer networks.



**Directed Graph** - relationships are 1-way.
Used to model airplane flights or bus routes.



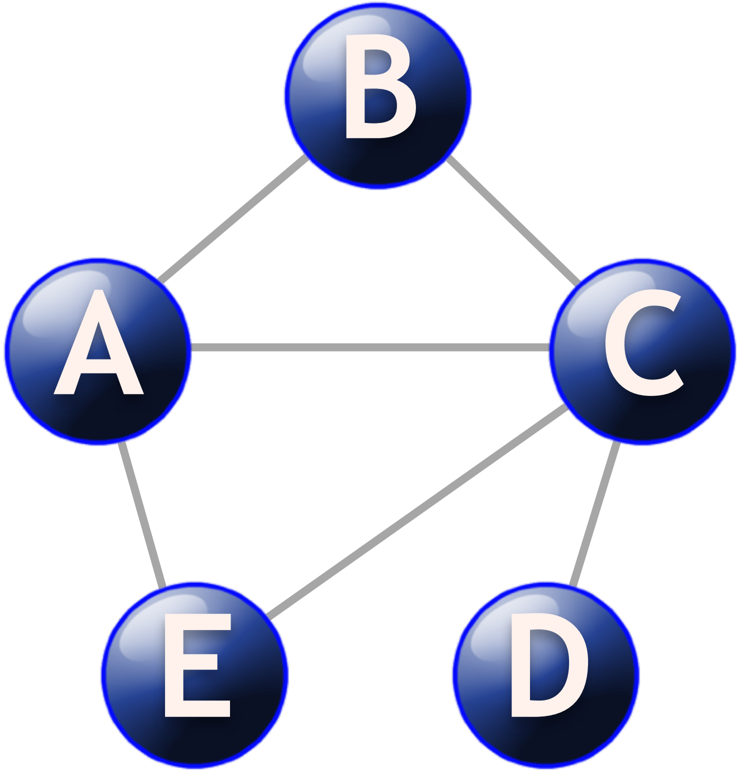
## Adjacency Matrix

Matrix of neighbors stored centrally in Graph object

## Adjacency List

List of neighbors stored in each vertex

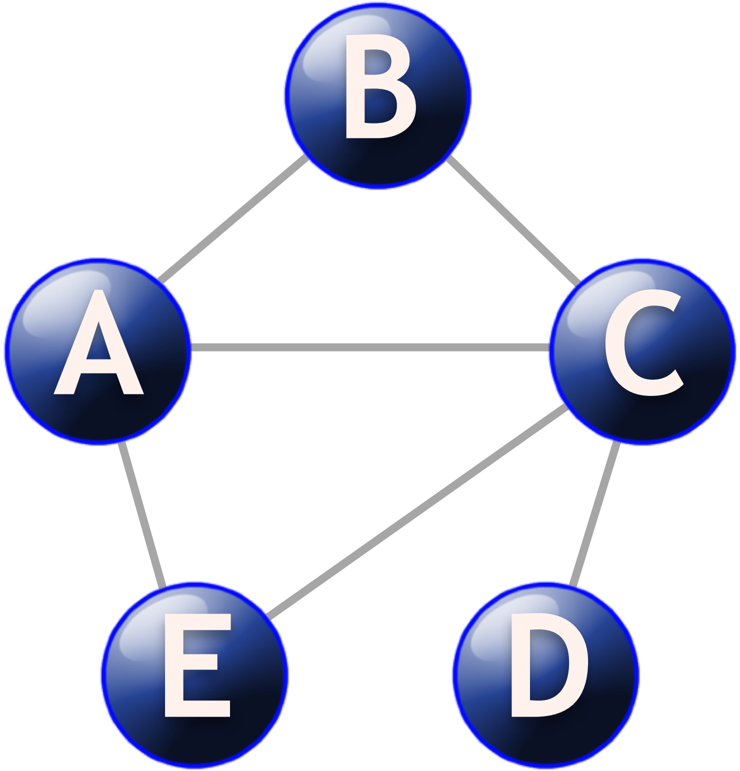
# Undirected Graph



## Adjacency List

A: B, C, E
B: A, C
C: A, B, D, E
D: C
E: A, C

# Undirected Graph



## Adjacency List

A: B, C, E

Stored in Node A

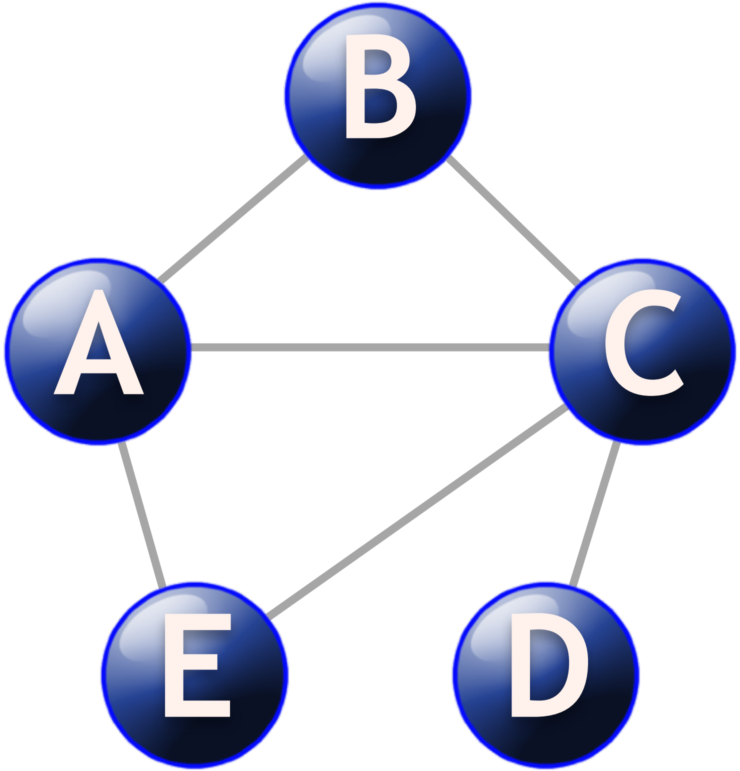
B: A, C

C: A, B, D, E

D: C

E: A, C

# Undirected Graph

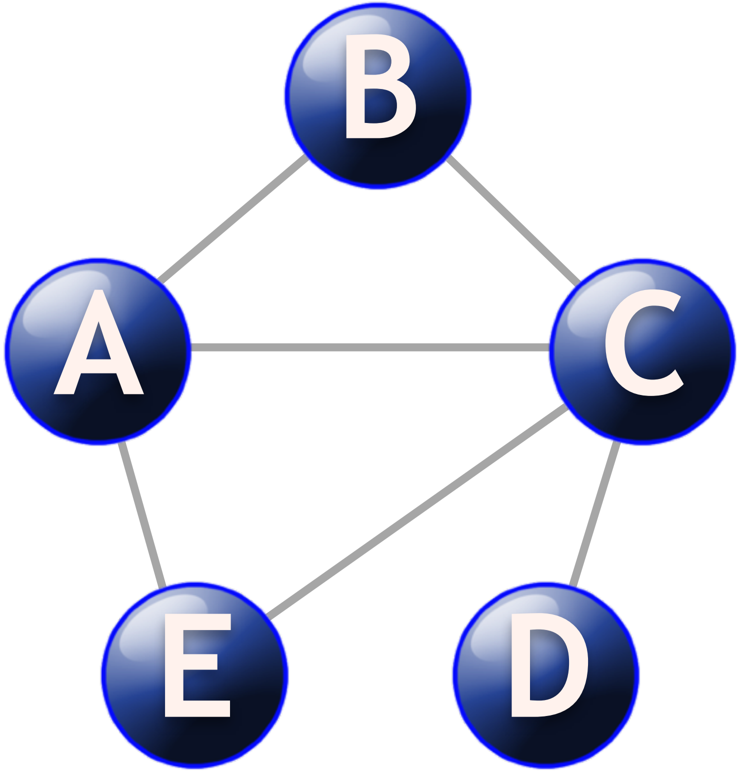


## Adjacency List

A: B, C, E
B: A, C
C: A, B, D, E
D: C
E: A, C

Stored in Node B

## Undirected Graph



## Adjacency Matrix

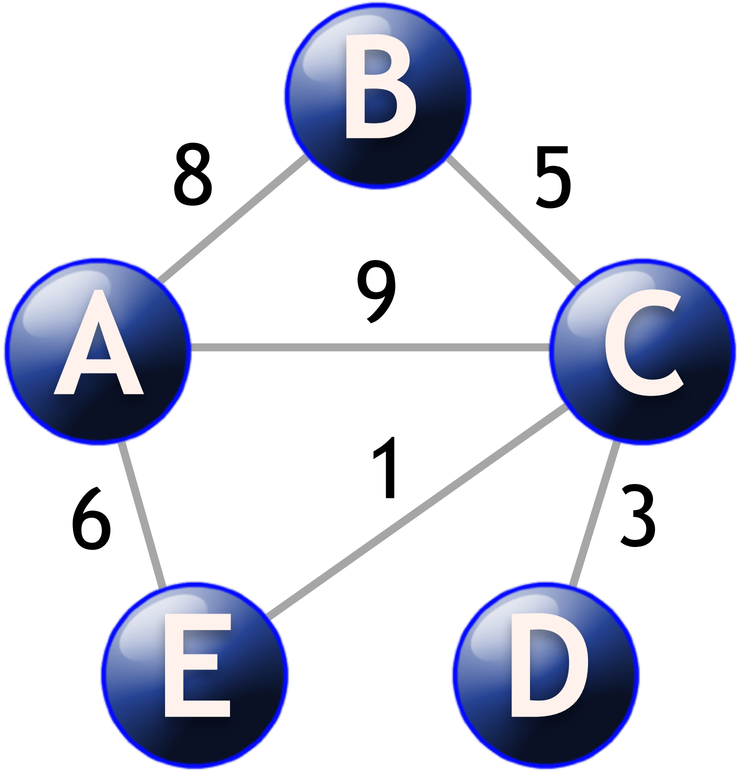
to

from

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 1 | 1 | 0 | 1 |
| B | 1 | 0 | 1 | 0 | 0 |
| C | 1 | 1 | 0 | 1 | 1 |
| D | 0 | 0 | 1 | 0 | 0 |
| E | 1 | 0 | 1 | 0 | 0 |

**Stored in Graph**

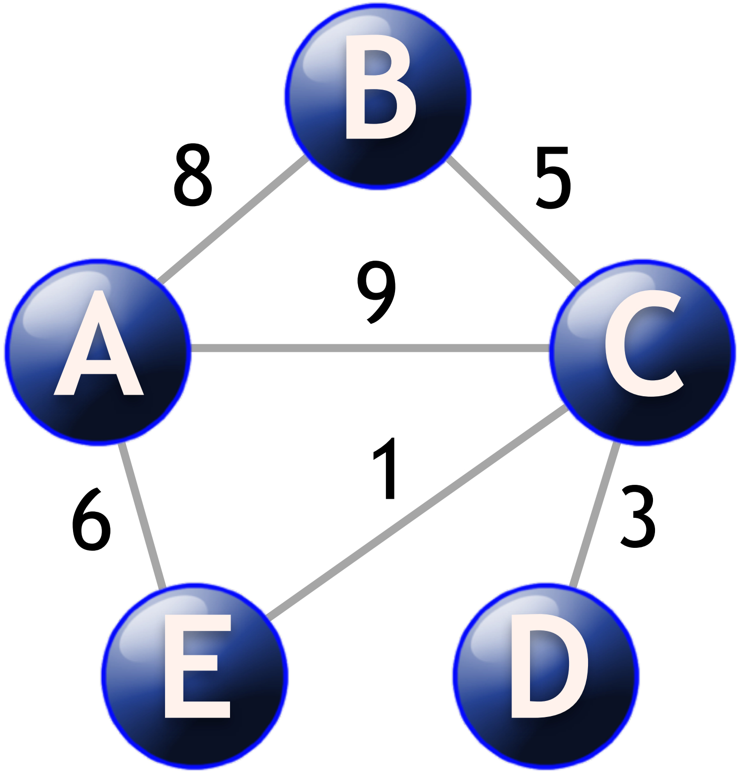
# Undirected Graph



## Weighted Edges?

Much easier to implement
with
Adjacency Matrix

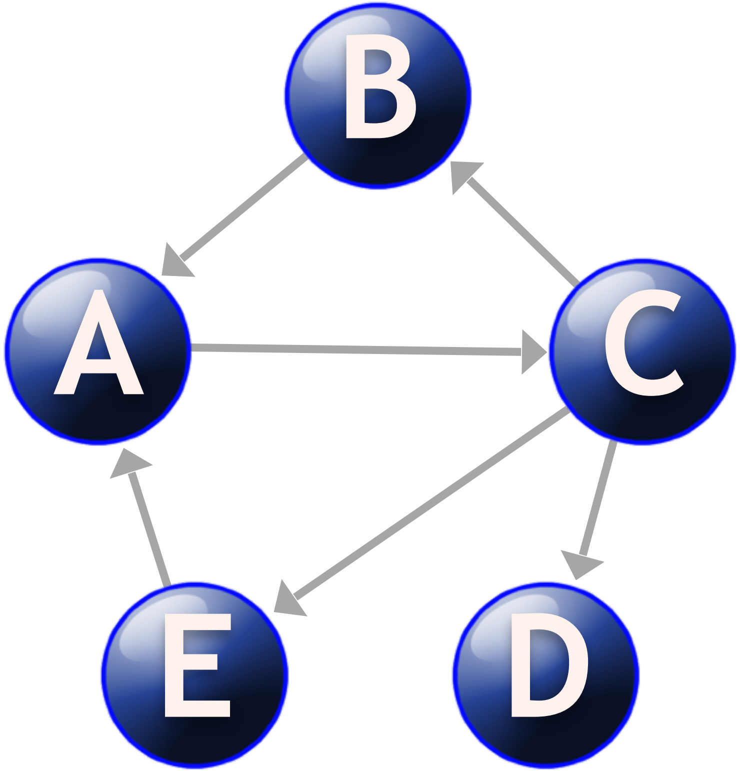
# Undirected Graph



# Adjacency Matrix

| | A | B | C | D | E |
|---|---|---|---|---|---|
| **A** | 0 | 8 | 9 | 0 | 6 |
| **B** | 8 | 0 | 5 | 0 | 0 |
| **C** | 9 | 5 | 0 | 3 | 1 |
| **D** | 0 | 0 | 3 | 0 | 0 |
| **E** | 6 | 0 | 1 | 0 | 0 |

# Directed Graph



## Adjacency List

A: C

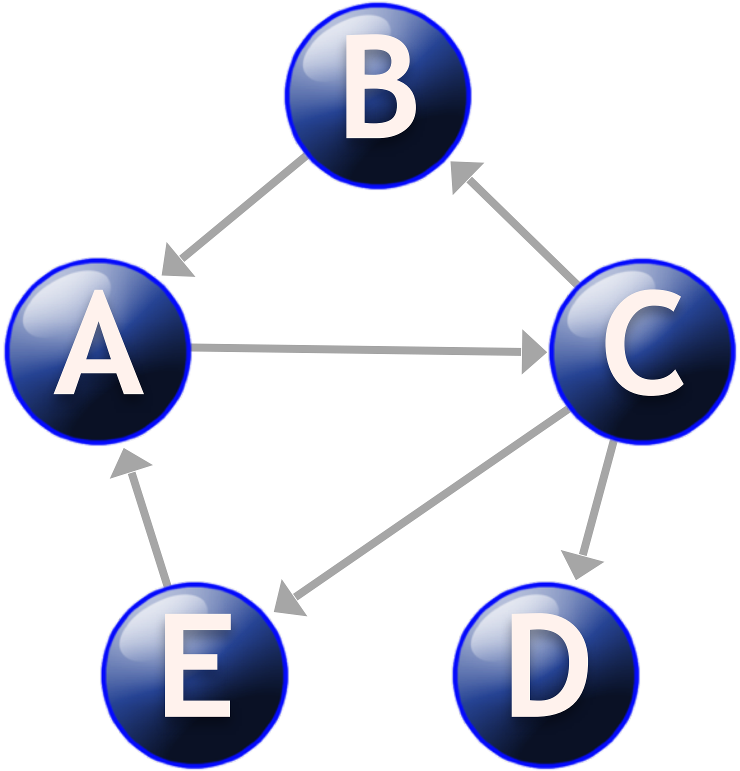
B: A

C: B, D, E

D:

E: A

# Directed Graph



# Adjacency Matrix

to

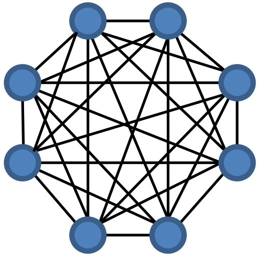
from

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 0 | 1 | 0 | 0 |
| B | 1 | 0 | 0 | 0 | 0 |
| C | 0 | 1 | 0 | 1 | 1 |
| D | 0 | 0 | 0 | 0 | 0 |
| E | 1 | 0 | 0 | 0 | 0 |

# Which is Better?

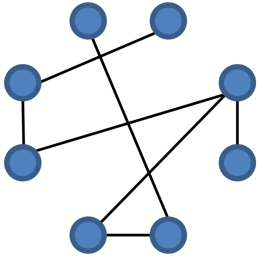
Dense Graph –

graph where $|E| = |V|^2$



Sparse Graph –

graph where $|E| = |V|$



# Which is Better?

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 0 | 1 | 0 | 0 |
| B | 1 | 0 | 0 | 0 | 0 |
| C | 0 | 1 | 0 | 1 | 1 |
| D | 0 | 0 | 0 | 0 | 0 |
| E | 1 | 0 | 0 | 0 | 0 |

Adjacency Matrix takes up $|V|^2$ space, regardless how dense the graph

Matrix for a graph with 10,000 vertices will take up at least 100,000,000 Bytes

# Which is Better?

**Adjacency List**

- Pro: Faster and uses less space for Sparse graphs
- Con: Slower for Dense graphs

**Adjacency Matrix**

- Pro: Faster for Dense graphs
- Pro: Simpler for Weighted edges
- Con: uses more space