COMP 2611, DATA STRUCTURES
LECTURE 17

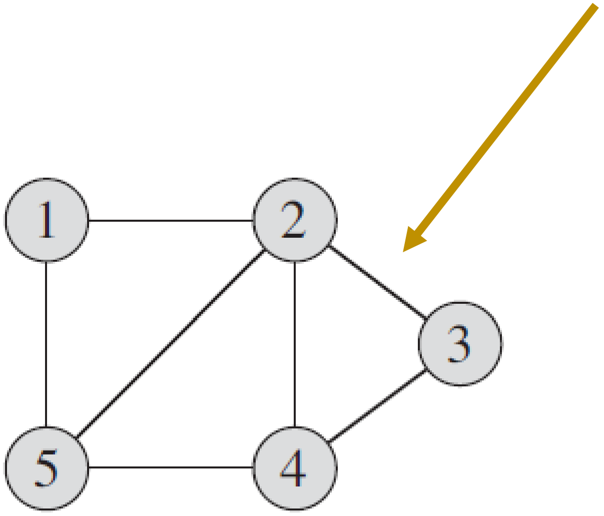
GRAPHS

- **Terminology**
- **Representation**
- **Building a Graph**
- **Weighted Graphs**

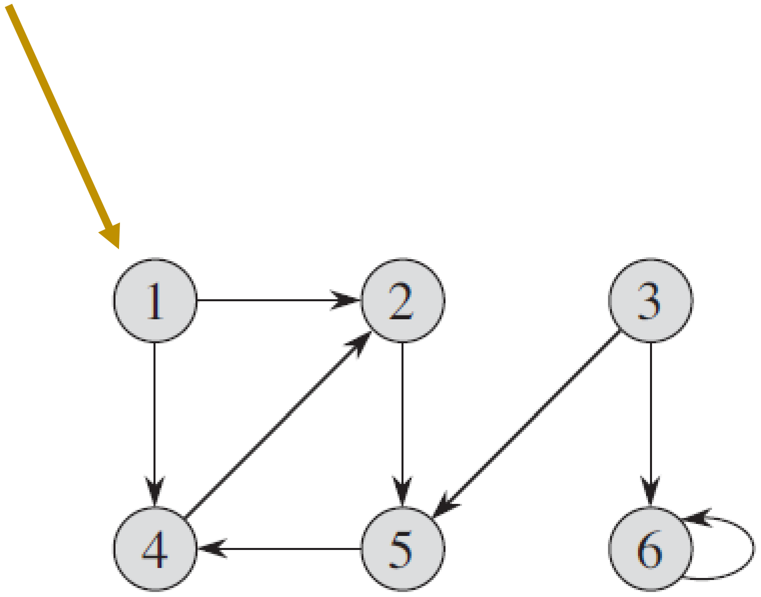
Graphs: Definitions

- A graph is a pair (*V*, *E*) where *V* is a finite set and *E* is a binary relation on *V*.

- Each element of *V* is called a *vertex* or *node* and each element of *E* is called an *edge*.



Undirected Graph



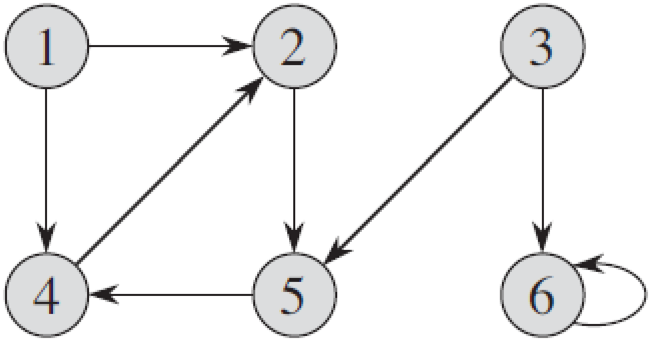
Directed Graph

Graphs: Terminology

- If a graph is directed, the *in-degree* of a vertex is the number of edges *entering* it.

- The out-degree of a vertex is the number of edges leaving it.

- What is the in-degree and out-degree of vertex 2?

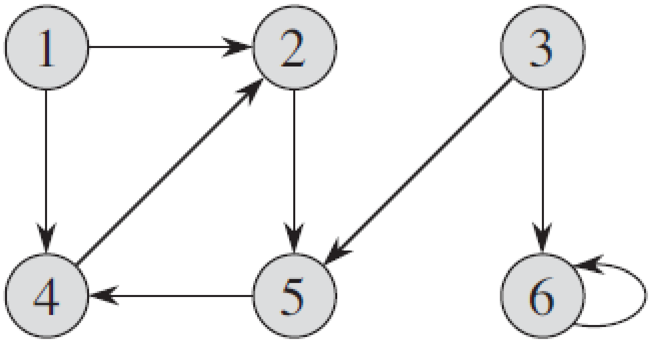


- The *degree* of a vertex is the sum of its in-degree and out-degree.

Graphs: Terminology

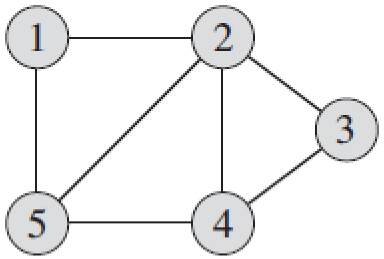
- If it is possible to get from vertex *u* to vertex *v* by following edges in the graph, we say that there is a *path* from vertex *u* to vertex *v*.

- The length of the path is the number of edges in the path.

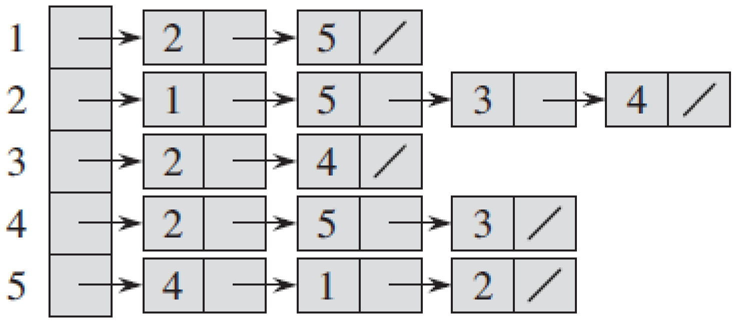


- A directed graph is *strongly connected* if there is at least one path between any two vertices.

Graphs: Representation



(a)



(b)

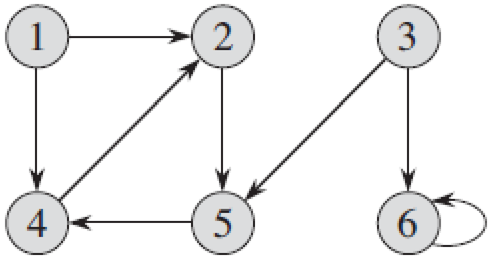
	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)

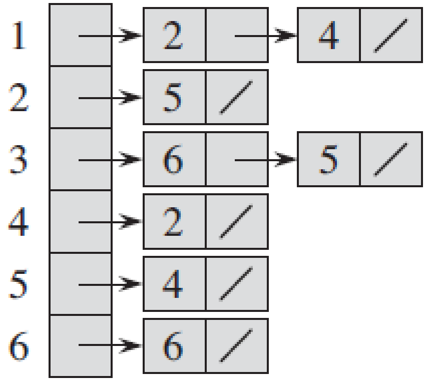
Adjacency List

Adjacency Matrix

Graphs: Representation



(a)



(b)

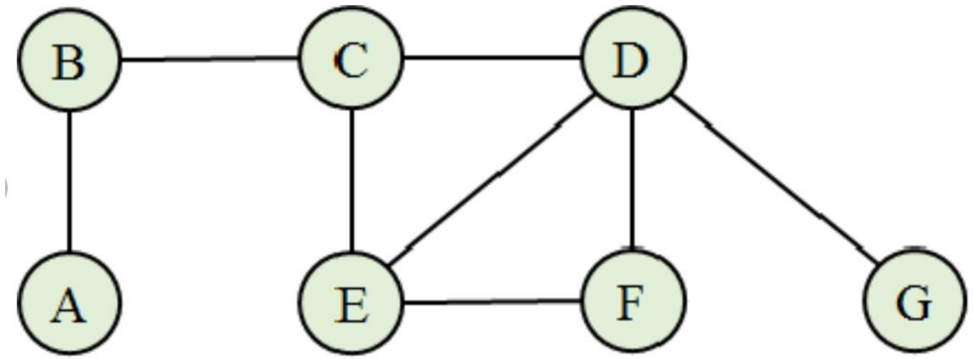
$$\begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 & 1 \\
4 & 0 & 1 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 1 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}$$

(c)

Adjacency List

Adjacency Matrix

Graphs: Adjacency Matrix Representation

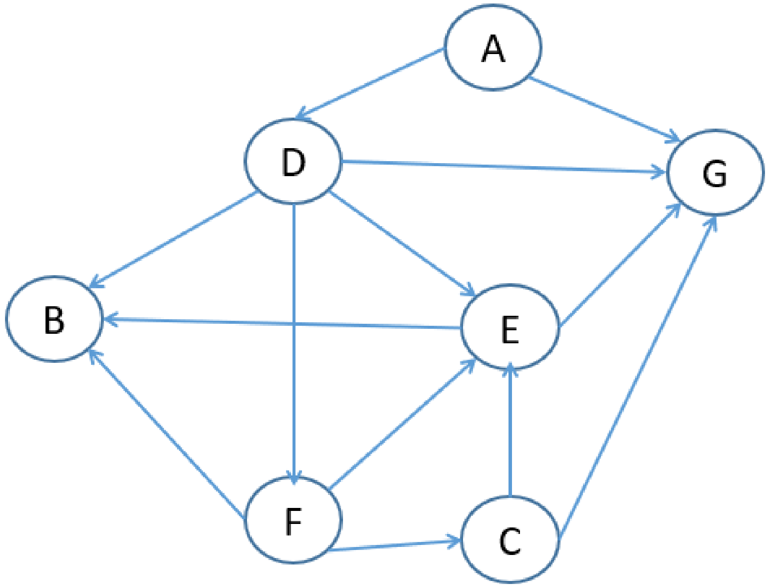


Graph

	A	B	C	D	E	F	G
A	0	1	0	0	0	0	0
B	1	0	1	0	0	0	0
C	0	1	0	1	1	0	0
D	0	0	1	0	1	1	1
E	0	0	1	1	0	1	0
F	0	0	0	1	1	0	0
G	0	0	0	1	0	0	0

Adjacency Matrix

Graphs: Adjacency List Representation



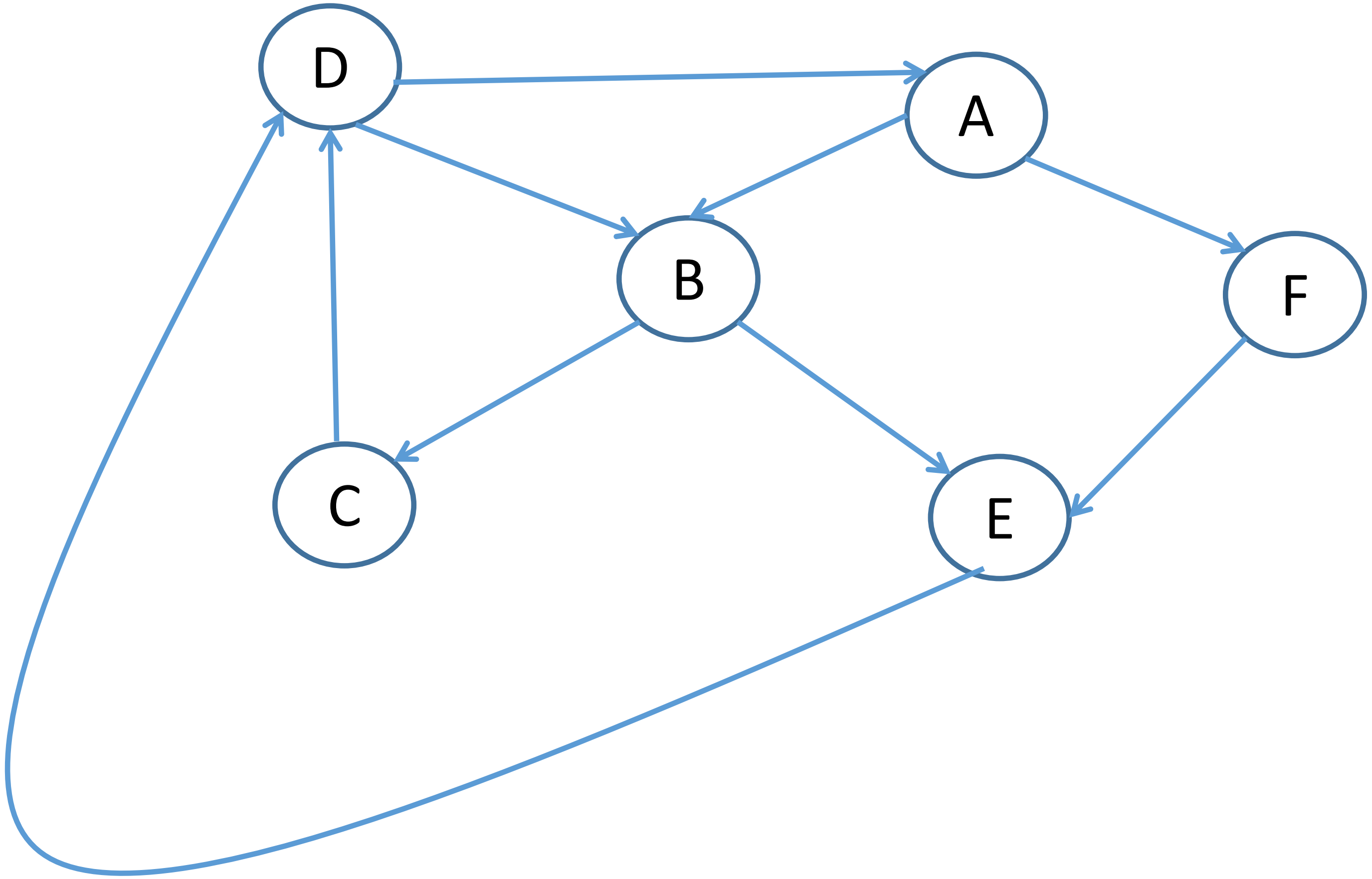
Graph

```
[A] -> D -> G
[B] ->
[C] -> E -> G
[D] -> B -> E -> F -> G
[E] -> B -> G
[F] -> B -> C -> E
[G] ->
```

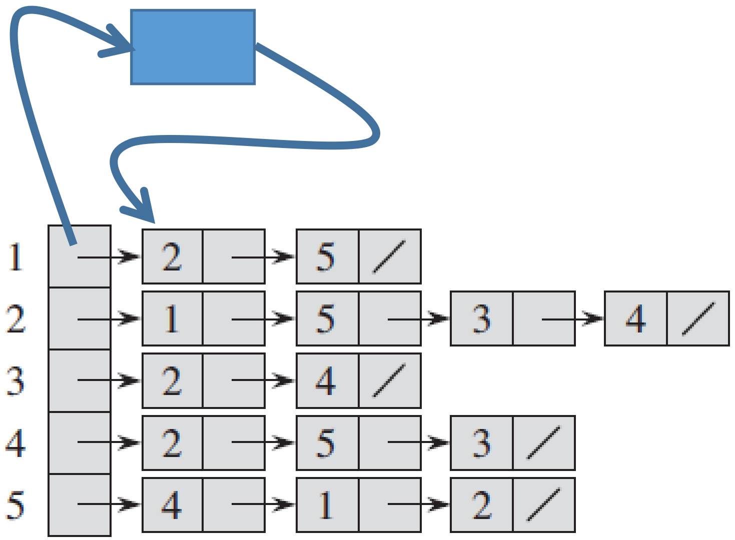
Adjacency List

Graphs: Storing in a File

```
6
A B C D E F
A 2 B F
B 2 C E
C 1 D
D 2 A B
E 1 D
F 1 E
```



Graphs: Data Structure



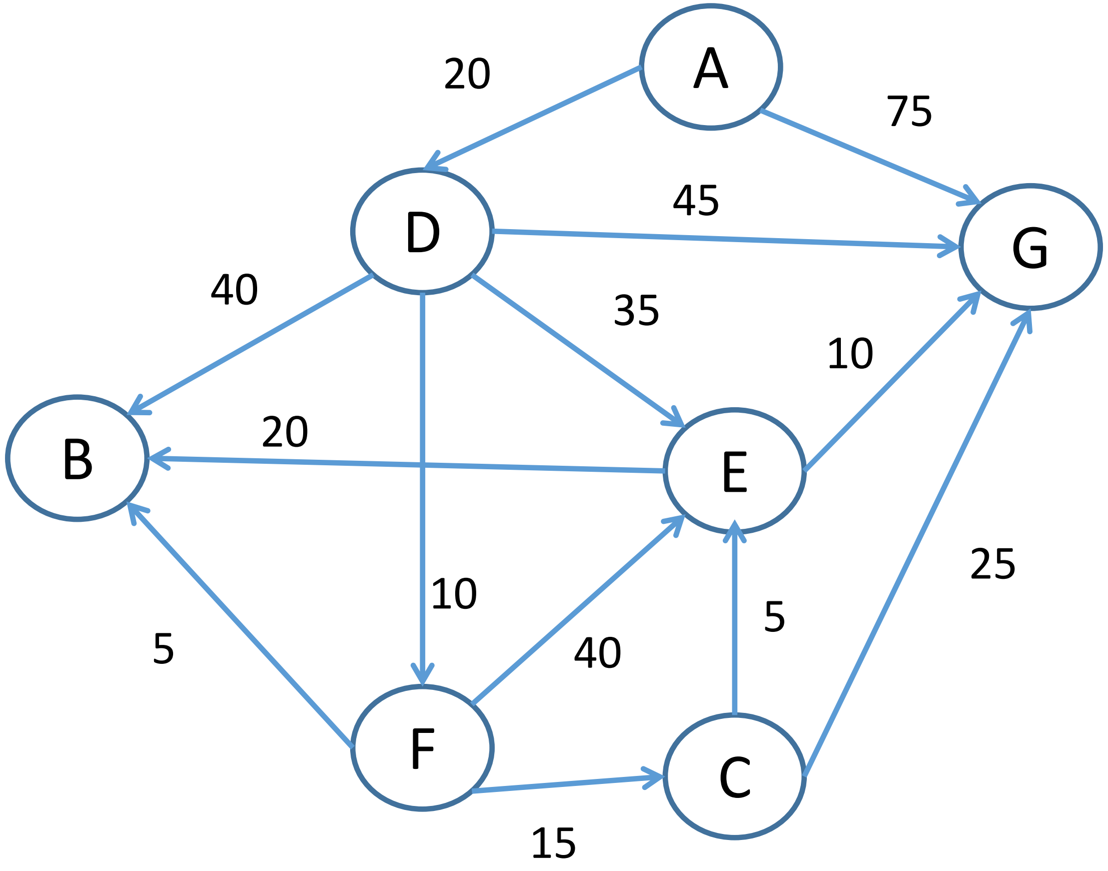
Adjacency List

```
struct Edge {
        string destID;
        Edge * nextEdge;
};

struct Vertex {
        string ID;
        Edge * firstEdge;
};

struct Graph {
        int numVertices;
        Vertex vertices[MAX_VERTICES];
};
```

Is there an edge from vertex *u* to vertex *v*?

What are the edges leaving vertex *v*?

Weighted Graphs


A
20
75
D
45
G
40
35
10
B
20
E
10
5
5
40
25
F
C
15