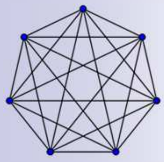


FLORIDA INTERNATIONAL UNIVERSITY

Graphs (an introduction)

Dr. Antonio L. Bajuelos

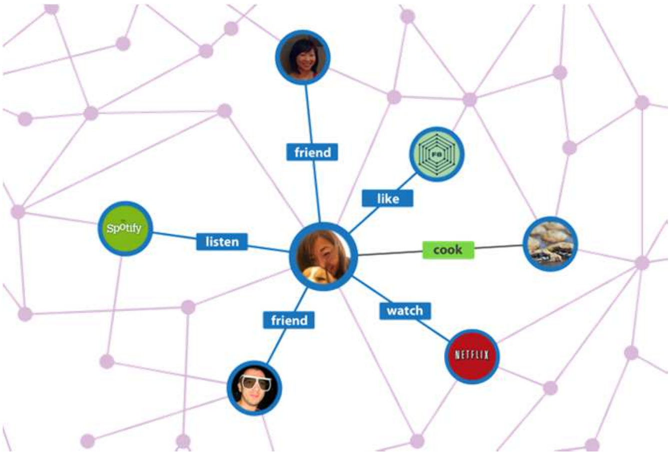
School of Computing & Information Sciences

Graphs. Some applications

- **Graphs** are **discrete structures** consisting of vertices/nodes and edges that connect these vertices.
- **Graphs** have many practical applications of computer science.
- For example:
  - In a **computer network**, we can model how the computers are connected to each other as a graph. The nodes are the individual computers and the edges are the network connections.
  - In a (digitalized) **map**, nodes are intersections (or cities), and edges are roads (or highways). We may have directed edges to capture one-way streets, and weighted edges to capture distance.
  - On the **internet**, nodes are web pages, and (directed) edges are links from one web page to another.
  - In a **social network**, nodes are people, and edges are friendships. Understanding social networks is a very hot topic of research. For example, how does a network achieve "six-degrees of separation", where everyone is approximately 6 friendships away from anyway else.
  - . . .

2

Graphs. The facebook graph



- From: http://o7planning.org/en/10189/exploring-facebook-graph-api.
- One Trillion Edges: Graph Processing at Facebook-Scale in:
  http://www.vldb.org/pvldb/vol8/p1804-ching.pdf

Graphs. Preliminary definitions

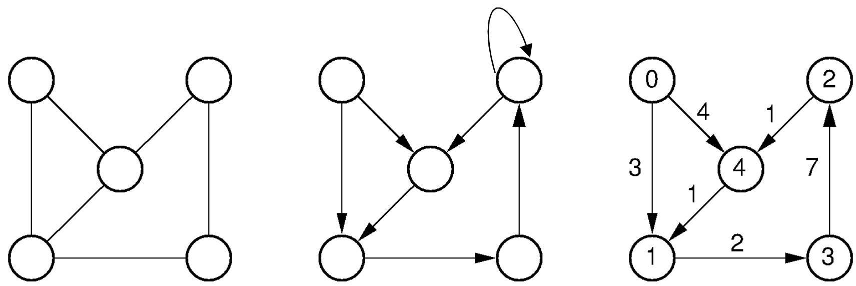
Definition:

- **A directed graph** G is a structure <V,E>, where V is a set of **vertices** (or **nodes**), and $E \subseteq V \times V$ is a set of **edges**.
- An **undirected graph** additionally has the property that $(u,v) \in E$ if and only if $(v,u) \in E$.

- **Note that:**
  - In **directed graphs**, edge (u,v) (starting from node u, ending at node v) is different from edge (v,u).
  - We also allow "self-loops" (**loops**), i.e., edges of the form (v,v) (say, a web page may link to itself).
  - In **undirected graphs**, because edge (u,v) and (v,u) must both be present or missing, we often treat a non-self-loop edge as an unordered set of two nodes (e.g., {u,v}).
  - A common extension is a **weighted graph**, where each edge additionally carries a weight (a real number). The weight can have a variety of meanings in practice: distance, importance, capacity, cost, etc…

Graphs. Preliminary definitions (cont…)

- Examples of **Graphs**:



Undirected Graph **Direct** Graph (**Digraph**) **Weighed** Graph

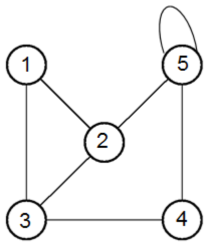
Graphs. Preliminary definitions (cont…)

- **Vertex Degree**: the degree of a vertex corresponds to the number of edges coming out or going into a vertex.

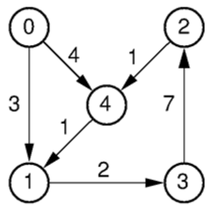
Definition:

- In a **directed graph** G = (V,E),
  - the **in-degree** (indegree) of a vertex v ∈ V is the number of edges coming in to it (i.e., of the form (u,v), u ∈ V);
  - the **out-degree** (outdegree) is the number of edges going out of it (i.e., of the form (v,u), u ∈ V).
  - The degree of v is the sum of the **in-degree** and the **out-degree**.
- In an **undirected graph**
  - the **degree** of v ∈ V is the number of edges going out of the vertex (i.e., of the form (v,u), u ∈ V), with the exception that self loops (i.e., the edge (v,v)) is counted twice.
- We denote the **degree** of vertex v ∈ V by **deg(v)**.

Graphs. Preliminary definitions (cont…)



- deg(1) = 2
- deg(2) = 3
- deg(5) = 4 = 2 + 2
- . . .



- indegree(1) = 2
- outdegree(1) = 1
- deg(1) = 3 = 2 + 1
- indegree(0) = 0
- outdegree(0) = 2
- deg(0) = 2 = 0 + 2
- . . .

Graphs. Preliminary results

Theorem:

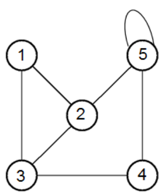
- Given a (directed or undirected) graph G = (V,E),

$$2|\mathrm{E}| = \sum_{v \in V} \deg(v)\,.$$

Corollary:

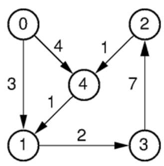
- In a graph, the number of vertices with an odd degree is even.

Examples:



|E| = 7; 2|E| = 14
deg(1) = 2
deg(2) = 3
deg(3) = 3
deg(4) = 2
deg(5) = 4

Σ = 14



deg(0) = 2
deg(1) = 3, **odd**
deg(2) = 2
deg(3) = 2
deg(4) = 3, **odd**

even

Graphs. Preliminary results (cont…)

Theorem:

- Given a (directed or undirected) graph G = <V,E>,

$$2|E| = \sum_{v \in V} \deg(v).$$

- **Exercise:** How many edges are there in a graph with 12 vertices each of degree five?
- **Solution:**
  - Because the sum of the degrees of the vertices is 5 x 12 = 60, it follows that 2|E| = 60
  - where |E| is the number of edges. Therefore, |E| = 30.

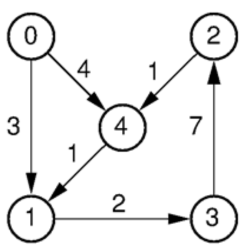
Graphs. Preliminary results (cont…)

Theorem:

- Given a directed graph G = (V,E),

$$|E| = \sum_{v \in V} outdegree(v) = \sum_{v \in V} indegree(v).$$

- **Example:**

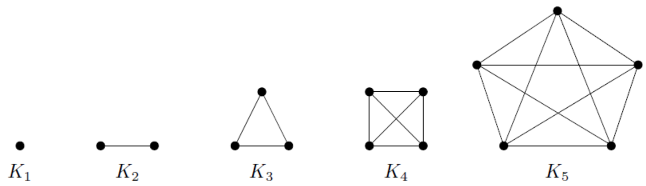


- **|E| = 6**
- indegree(0) = 0 outdegree(0) = 2
- indegree(1) = 2 outdegree(1) = 1
- indegree(2) = 1 outdegree(2) = 1
- indegree(3) = 1 outdegree(3) = 1
- indegree(4) = 2 outdegree(4) = 1

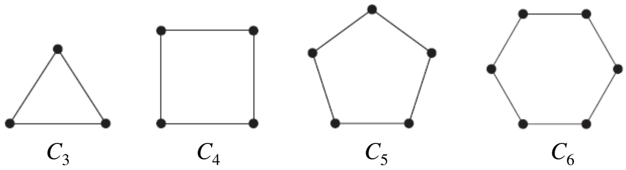
Σ = 6 **Σ = 6**

Some classes of simple graphs

- Simple graph – A graph with NO loops
- **Complete Graphs -** graph on n vertices, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices.

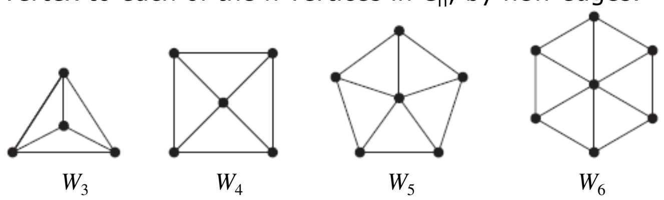


- **Cycle graphs** – a cycle C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $(v_1,v_2), (v_2,v_3), \ldots, (v_{n-1},v_n)$, and (v_n,v_1).

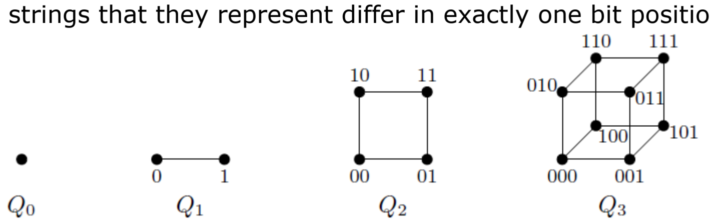


Some classes of simple graphs (cont…)

- **Wheel graphs** – we obtain a wheel W_n when we add an additional vertex to a cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges.

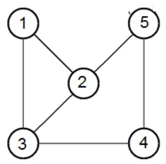


- **n-Cubes** - an n-dimensional hypercube, or n-cube, denoted by Q_n, is a graph that has vertices representing the 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position



Representing Graphs. Adjacency matrices

- Let a simple **graph** G = (V,E) and assume that |V|= n
- In the **adjacency matrix** representation, each graph of n nodes is represented by an n x n matrix A, that is, a two-dimensional array A
- The **nodes** are (re)-labeled 1, 2,…, n
  - A_{ij} = **1** if the edge (i,j) is an edge in the graph
  - A_{ij} = **0** if the edge (i,j) is **not** an edge in the graph
- Remember that, in a matrix, the first index represents the row-position, and the second the col-position.
- **Example:**

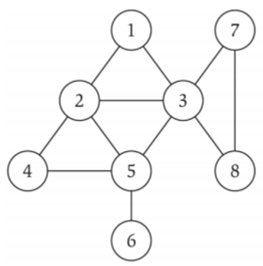


adjacency matrix

0 1 1 0 0
1 0 1 0 1
1 1 0 1 0
0 0 1 0 1
0 1 0 1 0

Representing Graphs. Adjacency matrices

- **Example:**



adjacency matrix

	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

- The **adjacency matrix** of a **simple undirected graph** is symmetric, that is, $a_{ij} = a_{ji}$, because both of these entries are 1 when v_i and v_j are adjacent, and both are 0 otherwise.
- Furthermore, because a simple graph has no loops, each entry a_{ii}, i = 1, 2, 3, . . . , n, is 0.

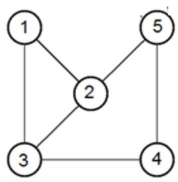
Representing Graphs. Adjacency matrices

- Adjacency matrix representation for **digraphs**??
- Adjacency matrix representation for **weighted digraph?**

15

Representing Graphs. Adjacency list

- Another way to represent a simple graph is to use **adjacency lists**, which specify the vertices that are adjacent to each vertex of the graph.
- **Example:**



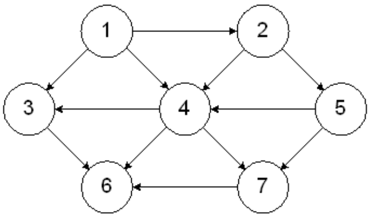
adjacency list

Vertex	Adjacent vertices
1	2, 3
2	1, 3, 5
3	1, 2, 4
4	3, 5
5	2, 4

16

Representing Graphs. Adjacency list (cont...)

- **Example (digraph):**



adjacency list (indegree)

Vertex	Adjacent vertices
1	- - -
2	1
3	1, 4
4	1, 2 ,5
5	2
6	3, 4, 7
7	4, 5

adjacency list (outdegree)

Vertex	Adjacent vertices
1	2, 3, 4
2	4, 5
3	6
4	3, 6, 7
5	4, 7
6	- - -
7	6

17