CS2123 Data Structures
Graphs

Dr. Weining Zhang

Department of Computer Science
University of Texas at San Antonio

October 7, 2009

Outline

Tasks of the Week

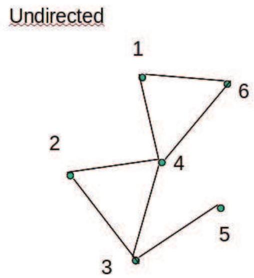
- Introduce undirected and directed graphs
- Learn basic graph algorithms
- Review for the final

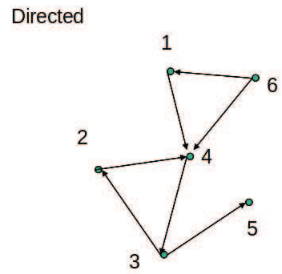
Graph

- Mathematically, a graph G has a set of vertexes $V(G) = \{1, 2, 3, \ldots, N\}$ and a set of edges $E(G) = \{(u, v) | u, v \in V(G)\}$
- Undirected Graph: edges have no direction $(u, v) \Leftrightarrow (v, u)$
- Directed Graph: edges have directions (u, v) does not imply (v, u)

Example Graphs

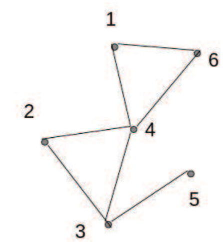
Vertexes: $V(G) = \{1,2,3,4,5,6\}$
Edges: $E(G) = \{(1,4),(1,6),(2,3),(2,4),(3,4),(3,5),(4,6)\}$





Terminology

- Degree: # of edges of a node
  - Ex: deg(1) = 2, deg(5) = 1, deg(4) = 4
  - In-degree, out-degree
- Adjacent nodes: linked by an edge
  - Ex: 1 and 4, 2 and 3
- Incident
  - Ex: 2 is incident of edge (2, 3)



Graph Properties

Example

In an undirected graph,

$$|E(G)| \leq \frac{|V(G)| \times (|V(G)| - 1)}{2}$$

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

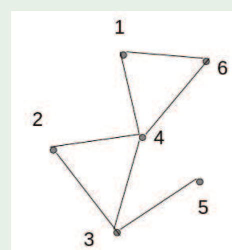
- There are many such properties. For details, read books on Graph Theory.

Representations

- Adjacency Matrix

Example

Undirected Graph



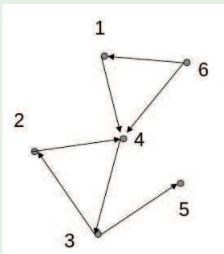
$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Representations

- Adjacency Matrix

Example

Directed Graph



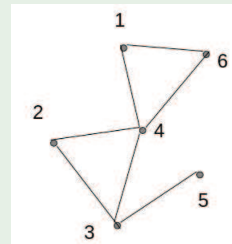
$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

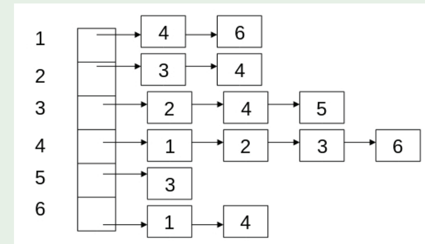
Representations

- Adjacency Lists

Example

Undirected Graph





Graph Algorithms

- Graph Traversals: visit every vertex once
  - Breadth-first
  - Depth-first
- Connectivity
  - A graph is connected if and only if the breadth-first traversal visits $|V(G)|$ vertexes
- Shortest paths
  - Assume each edge has a weight. find a minimum weight path between
    - a single pair of vertexes
    - between all pairs of vertexes