CS301
Data Structure and Algorithms
Lecture 9: Introduction to graphs

Pandav Patel
Assistant Professor

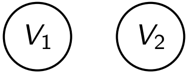
Computer Engineering Department
Dharmsinh Desai University
Nadiad, Gujarat, India

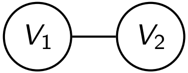
Objective

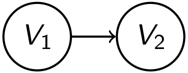
- To introduce graph data structure
- To know different types of graphs
- To get familiarized with graph terminologies

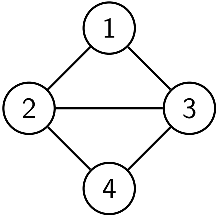
Overview

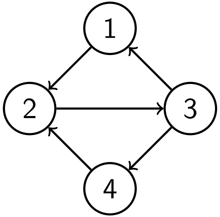
Visualization of graphs

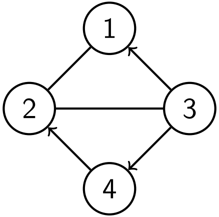












Visualization of graphs (cont...)

- **Vertices** (a.k.a. **nodes, points**) are labeled V_1, V_2, V_3... or 1, 2, 3...
- Two nodes can be connected by an **edge** and are called **adjacent nodes**
- Graph must have at least one vertex
- Graph can have zero or more edges
- First graph is the simplest graph with one vertex and no edges
- Second graph consists two vertices without any edges
- Edges can be **directed** (with direction) or **undirected** (no specific direction)
- NOTE: Adjacency in case of directed edge is defined differently by various authors
- **Directed graph:** Every edge is directed
- **Directed graph:** Every edge is undirected
- **Mixed graph:** Some edges are directed and some are undirected

Examples of graphs

- Consider a graph where nodes (vertices) represent intersections of the city and edges represent streets connecting the intersections
  - Directed graph: A city map showing only one-way streets
  - Undirected graph: A city map showing only two-way streets
  - Mixed graph: A city map showing one-way and two-way streets

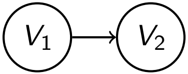
- Is graph linear data structure? Why?

Definition

- A *graph* G consists of a nonempty set V called the set of *nodes* (a.k.a. *points, vertices*) of the graph, a set E which is the set of *edges* of the graph, and a mapping from the set of edges E to a set of pairs of elements of V.

Loop (sling)

- In below graph, an edge is *initiating* or *originating* in the node V_1 and *terminating* or *ending* in the node V_2
- Node V_1 is called *initial* node and node V_2 is called *terminal* node for a given edge

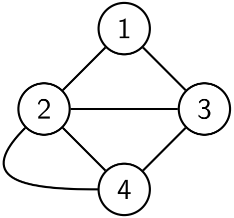


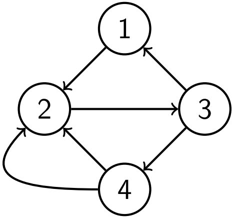
- An edge of the node which joins itself is called a *loop* (*sling*)
- Initial and terminal node is same for a loop

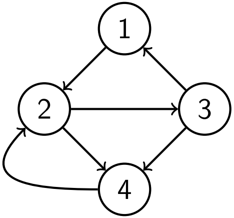
- Direction of the loop has no significance; hence, it can be considered either a directed or an undirected edge.

Parallel Edges

- If two nodes are joined by more than one edge; such edges are called *parallel*.
  - In case of directed graph, an edge from node A to B and an edge from node B to A are considered distinct. They are not considered parallel.



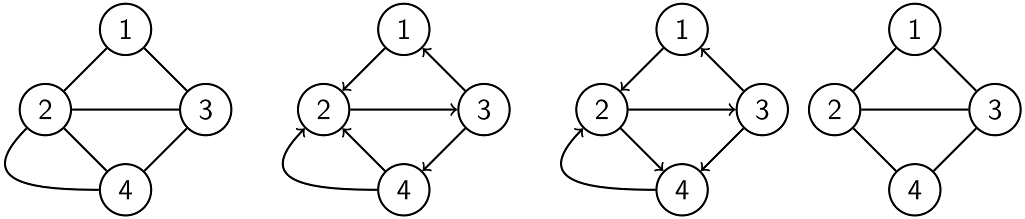




- There are parallel edges between node 2 and node 4 in first two graphs
- There are no parallel edges in third graph. Edges between nodes 2 and 4 in third graph are opposite in direction and hence are not considered parallel.

Multigraph Vs Simple Graph

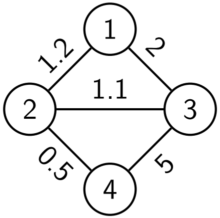
- If a graph contains any parallel edges then it is called *multigraph*.
- If a graph does not contain any parallel edges then it is called *simple graph*.

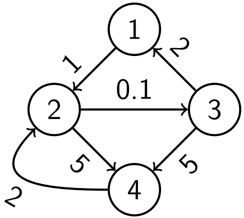


- First and second graphs are *multigraphs*
- Third and fourth graphs are *simple graphs*

Weighted graph

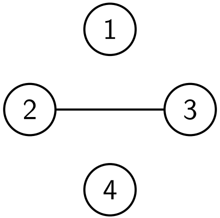
- A graph in which weights are assigned to every edge is called a *weighted graph*
  - Consider a graph where nodes represent intersections of the city and edges represent streets connecting the intersection. Weights can be assigned to each edge to according to
    - distance between two intersections joined by an edge
    - or according to traffic on the road represented by particular edge

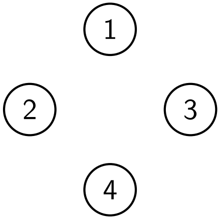




Isolated node and null graph

- A node which is not adjacent to any other node is called *isolated node*
- A graph containing only isolated nodes is called *null graph*
    - Set of edges in null graph will be empty

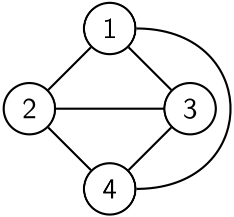


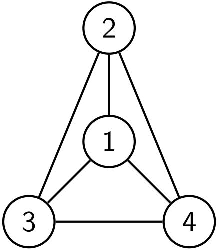


- First graph has two isolated nodes - 1 and 4. But it is not a null graph
- All nodes in a second graph are isolated hence it is a null graph

Same graph with different visualization

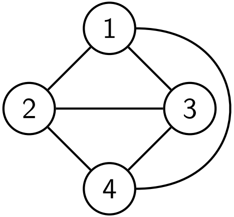
- Are these graphs same or different?

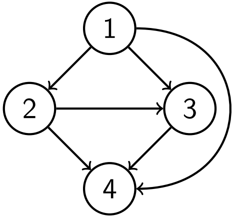




Degree of a node

- In a directed graph
  - For any node V the number of edges which have V as their
    - initial node is called *outdegree* of V
    - terminal node is called *indegree* of V
  - Sum of *outdegree* and *indegree* of node V is called *total degree* of node V
- In an undirected graph
  - *Total degree* or *degree* of node V is equal to number of edges incident with node V

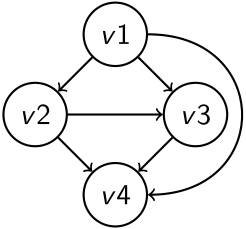




- In first graph, degree of all nodes is 3
- In second graph, node 1 has indegree 0 and outdegree 3; while node 2 has indegree 1 and outdegree 2
- Total degree of a loop is 2 and that of an isolated node is 0

Path

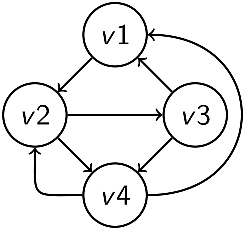
- *Path* is a sequence of edges of a graph such that the terminal node of any edge in the sequence is the initial node of the next edge, if any, in the sequence
  - A path is said to *traverse* through the nodes appearing in the sequence, *originating* in the initial node of the first edge and *ending* in the terminal node of the last edge in the sequence
- Number of edges appearing in the path is called *length* of the path



- Path ((v1, v3), (v3, v4)) traverses through nodes v1, v3, and v4. It originates in node v1 and ends in node v4.
  - Sometimes it is represented as (v1, v3, v4)
  - Its length is 2
- Is there any path of length 3 in the given graph? If yes, how many?

Simple Path and Elementary Path

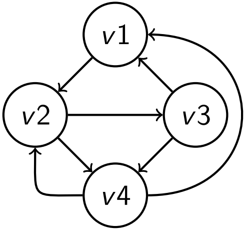
- A path in which edges are distinct is called an *simple path (edge simple)*
- A path in which all the nodes through which it traverses are distinct is called an *elementary path (node simple)*



- ((v1, v2), (v2, v3), (v3, v4)) is a simple (and elementary) path
- (v3, v4, v2, v4, v1) is an simple (but not elementary) path
- (v3, v4, v2, v4, v2, v3) is neither elementary nor simple path
- Every elementary path is also a simple path. But a simple path may or may not be elementary
- If there exists a path from node X to Y; then there must exist an elementary path from node X to Y. **Think through this**

Cycle (Circuit)

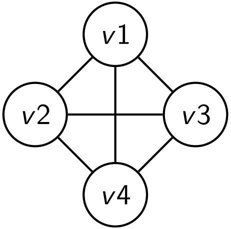
- A path which originates and ends in the same node is called a *cycle (circuit)*
- A cycle is called *elementary cycle* if it does not traverse through any node more than once (ignore the ends)

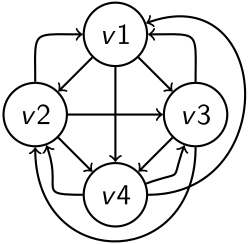


- (v3, v4, v2, v4, v2, v3) is a cycle (but it is not elementary cycle)
- (v3, v4, v2, v3) is an elementary cycle
- It is possible to obtain elementary cycle at any node from a cycle at that node **Think through this**
- A simple digraph which does not have any cycles is called *acyclic*

Complete graph

- A simple *undirected* graph in which every pair of distinct vertices is connected by a unique edge is called *complete graph*
- A simple *directed* graph in which every pair of distinct vertices is connected by a pair of unique edges (one in each direction) is called *complete graph*





- How many edges would be present in undirected complete graph of *n* nodes (ignore loops)?
- How many edges would be present in complete digraph of *n* nodes (ignore loops)?