# CSCI 2270 – Data Structures

Recitation 9, October 2020

## Graph Algorithms

## Objectives

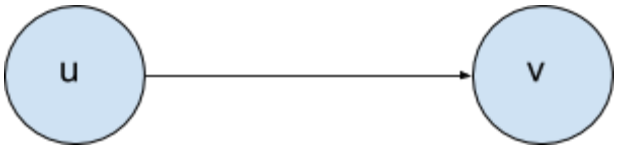
1. What is a Graph?
2. Exercise
    a. Challenge 1 (Silver Badge)

In the real world, many problems are represented in terms of objects and connections between them. For example, in an airline route map, we might be interested in questions like: “What’s the fastest way to go from Denver to San Francisco” or “What is the cheapest way to go from Denver to San Francisco” To answer these questions we need information about connections between objects. Graphs are data structures used for solving these kinds of problems.

Graph: A graph is a pair (V, E), where V is a set of nodes, called vertices and E is a collection of pairs of vertices, called edges.

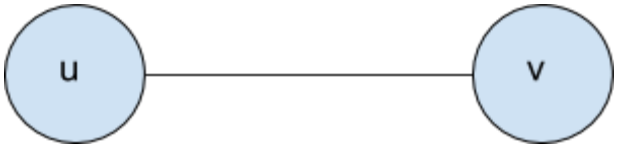
### Directed Edge

- Ordered pair of vertices (u, v)
- First vertex u is the origin
- Second vertex v is the destination
- Example: one way road traffic




### Undirected Edge:

- Unordered pair of vertices (u, v)
- Example: Railway lines



# CSCI 2270 – Data Structures

Recitation 9, October 2020

# Graph Algorithms

## Applications of Graphs

- Representing relationships between components in electronic circuits.
- Transportation networks: Highway network, Flight network
- Computer networks: Local area network, Internet web

## Graph Representation

As in other Abstract Data types, to manipulate graphs we need to represent them in some useful form. Basically, there are two ways of doing this:

- Adjacency Matrix
- Adjacency Lists

## Graph Traversals

To solve problems on graphs, we need a mechanism for traversing the graphs. Graph traversal algorithms are also called graph search algorithms. Like tree traversal algorithms (Inorder, Preorder, Postorder traversals), graph search algorithms can be thought of as starting at some source vertex in a graph, and 'search' the graph by going through the edges and making the vertices. Next week we will discuss two such algorithms for traversing the graphs.

- Depth First Search (DFS)
- Breadth First Search (BFS)

## Exercise

### A. Silver Badge Problem (Mandatory)

1. Download the Lab9 **zipped** file.
2. Follow the TODOs and complete the printGraph() function which prints all the graph vertices with their adjacency list.