INTRODUCTION TO GRAPHS

Section Overview

- To learn what a graph is and how it is used.
- To implement the **graph** abstract data type using multiple internal representations.
- To see how graphs can be used to solve a wide variety of problems

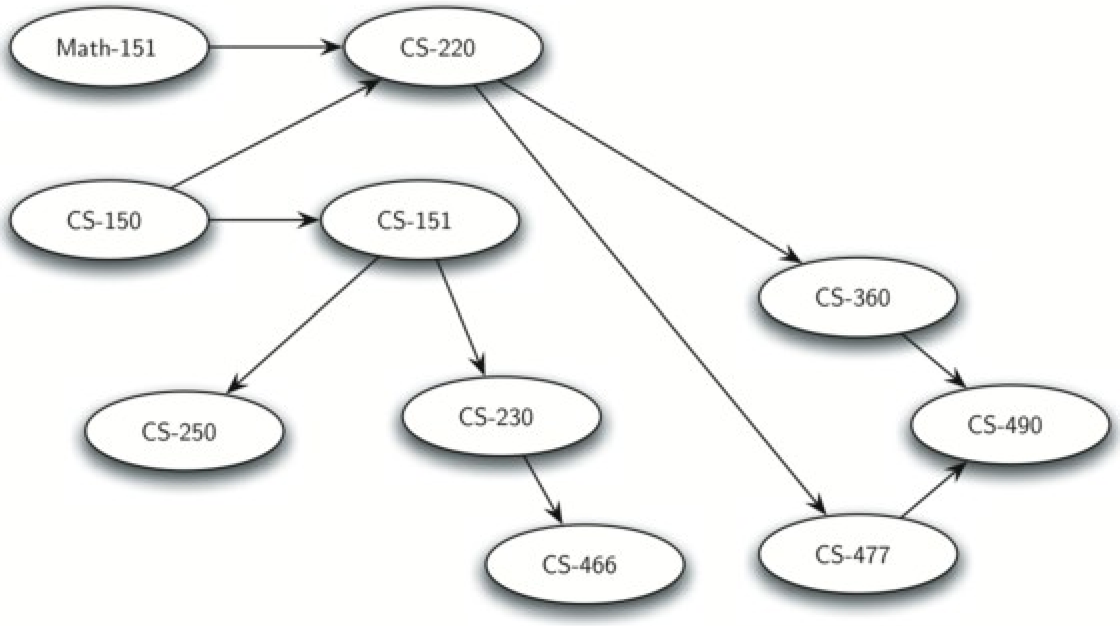
Section Overview

- Graphs are a more general structure than trees we can think of a tree as a special kind of graph.
- Graphs can be used to represent many real-world things such as systems of roads, airline flights from city to city, how the Internet is connected, etc.
- Once we have a good representation for a problem, we can use some standard graph algorithms to solve what otherwise might seem to be a very difficult problem.

Section Overview

- Computers can operate well with information presented as a graph.
- An example graph may be the course requirements for a computer science major

Example Graph



Vocabulary and Definitions

- Now that we have looked at some examples of graphs, we will more formally define a graph and its components.
- We already know some of these terms from our discussion of trees.

Vertex (Nodes)

- A **vertex** (also called a “**node**”) is a fundamental part of a graph.
- It can have a name, which we will call the “key.”
- A vertex may also have additional information.
- We will call this additional information the “payload.”

Edge

- An edge connects two vertices to show that there is a relationship between them.
- Edges may be one-way or two-way.
- If the edges in a graph are all one-way, we say that the graph is a **directed graph**, or a **digraph**.
- The class prerequisites graph shown previously is clearly a digraph since you must take some classes before others.

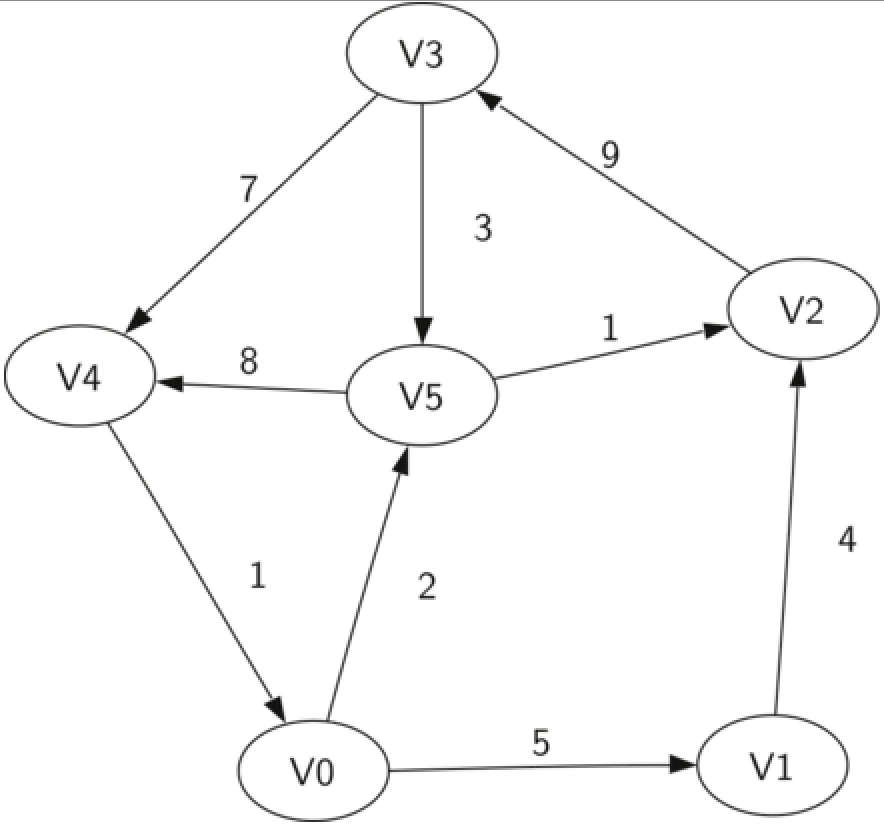
Weight

- Edges may be weighted to show that there is a cost to go from one vertex to another.
- For example in a graph of roads that connect one city to another, the weight on the edge might represent the distance between the two cities.

Formal Definition of a Graph

- A graph can be represented by **G** where **G=(V,E)**
- For the graph **G**, **V** is a set of vertices and **E** is a set of edges.
- Each edge is a tuple **(v,w)** where **w,v∈V**
- We can add a third component to the edge tuple to represent a weight.
- A subgraph **s** is a set of edges **e** and vertices **v** such that **e⊂E** and **v⊂V**

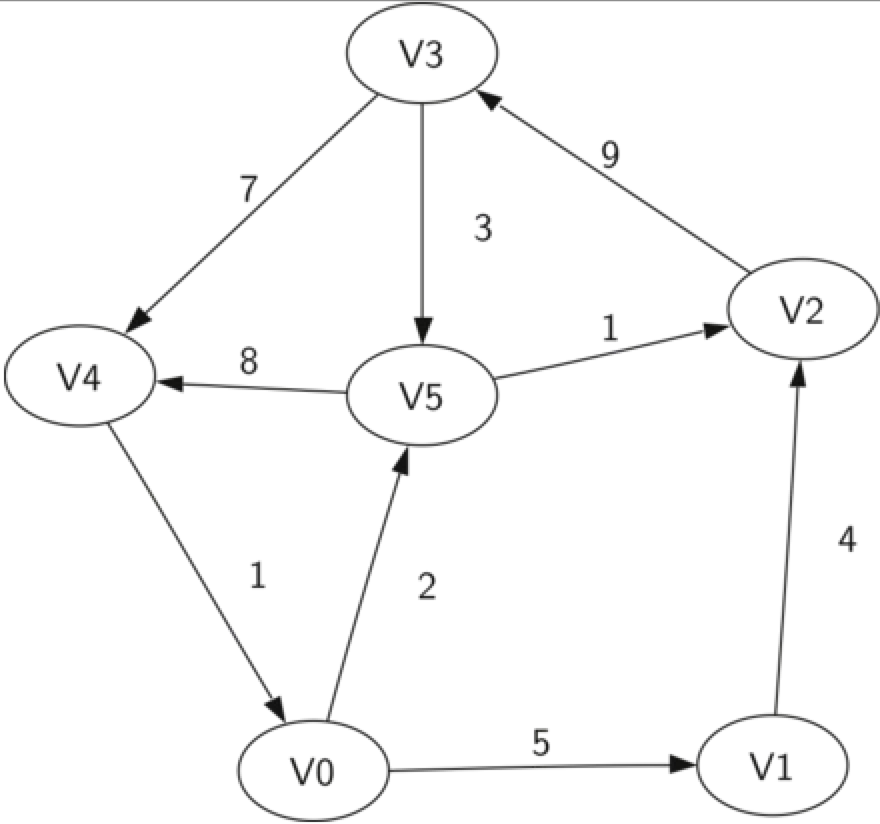
Example



Example

- V={V0,V1,V2,V3,V4,V5}

- E={(v0,v1,5),(v1,v2,4),
  (v2,v3,9),(v3,v4,7),
  (v4,v0,1),(v0,v5,2),
  (v5,v4,8),(v3,v5,3),
  (v5,v2,1)}

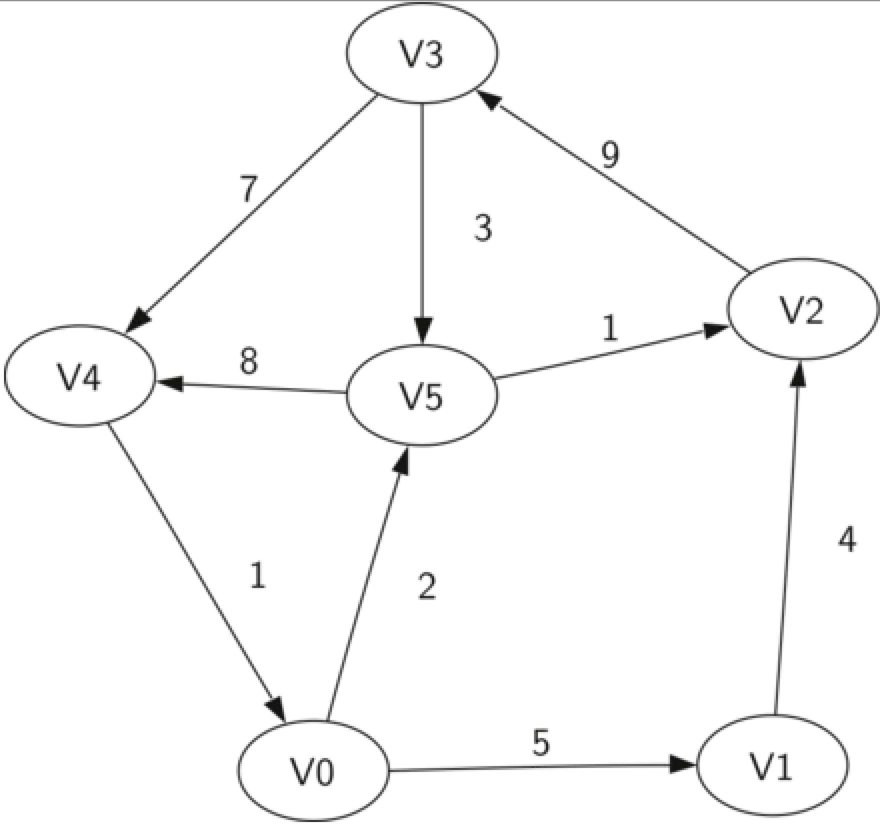


Path

- A path in a graph is a sequence of vertices that are connected by edges.
- Formally we would define a path as $\mathbf{w_1, w_2, \ldots, w_n}$ such that $\mathbf{(w_i, w_{i+1}) \in E}$ for all $\mathbf{1 \leq i \leq n-1}$
- The unweighted path length is the number of edges in the path, specifically $\mathbf{n-1}$.
- The weighted path length is the sum of the weights of all the edges in the path.

Path Example

- The path from **V3** to **V1** is the sequence of vertices **(V3,V4,V0,V1)**
- The edges are **{(v3,v4,7),(v4,v0,1),(v0,v1,5)}**

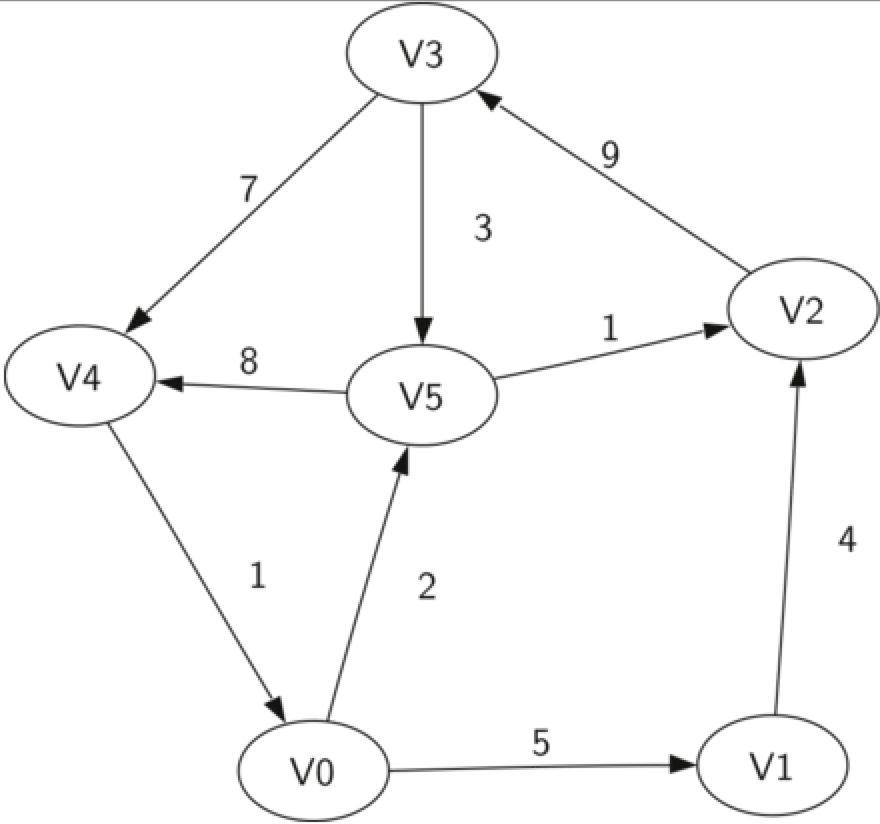


Cycle

- A cycle in a directed graph is a path that starts and ends at the same vertex.
- A graph with no cycles is called an **acyclic graph**.
- A directed graph with no cycles is called a **directed acyclic graph** or a **DAG**.
- We will see that we can solve several important problems if the problem can be represented as a DAG.

Cycle Example

- The path **(V5,V2,V3,V5)** is a cycle.



Review

- Definition of a Graph
- Important Graph Vocabulary Terms
- Up next – how to represent and implement a Graph.