Graphs

Henry Z. Lo

July 4, 2014

1 Graphs

1.1 Definition

Graphs are a combination of *vertices* (or nodes) and *edges*. Edges connect vertices. See an example in Figure **??**.

Graphs can have *directed* or *undirected* edges. Figure **??** shows an undirected graph. Edges here represent symmetric friendships, and so do not need to have direction. Figure **??** shows the water cycle in a directed graph. Here, directed edges makes sense because water does not evaporate and then become groundwater.

We can define the following on a graph:

- **Path**: a path between two nodes is a sequence of edges connecting them.
- **Cycle**: a path which begins and ends at the same vertex.
- **Connected component**: a set of nodes which have paths to each other.

Edges can have (and often do) weights, or numerical values assigned to them. They can represent anything from strength of friendship, time it takes on a road, etc.

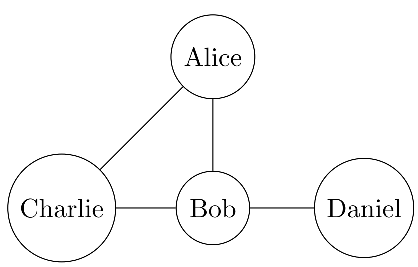

Figure 1: Example of a graph representing friendships.

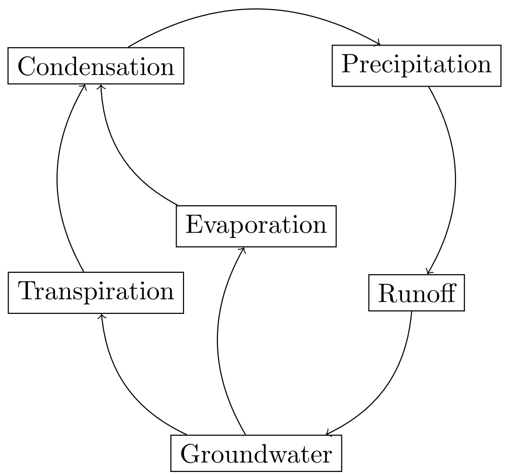

Figure 2: Directed graph representing the water cycle.

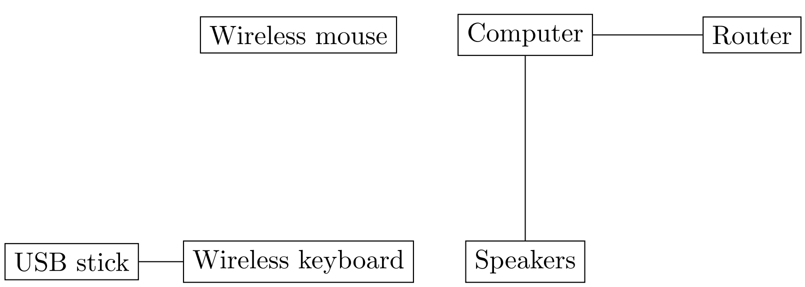

Figure 3: Example of a graph with three connected components.

1.2 Relationship with trees

A tree is a type of graph which has the following properties:

- It has one connected component.
- It does not have cycles with more than 2 nodes.
- We select some node as a root.

Because of this, we can define a *spanning tree* on a graph, which contains the same number of nodes, but with no cycles. This type of structure can be organized into a tree. There are many *spanning trees* for a graph.

2 Representations

2.0.1 Adjacency list

2.0.2 Adjacency matrix

2.0.3 Object representation

3 Minimum Spanning Trees

Suppose you are a telecom company, and you need to lay cable. You need to hit each home (node), but taking different paths may result in different costs, e.g. you may need to dig deeper in some areas.

3.1 Prim's algorithm

1. Initialize a tree with a single vertex, chosen arbitrarily from the graph.
2. Grow the tree by one edge: of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge, and transfer it to the tree.
3. Repeat step 2 (until all vertices are in the tree).

$O(V^2)$ adjacency matrix

`img/matrix.png` `img/graph1.png`