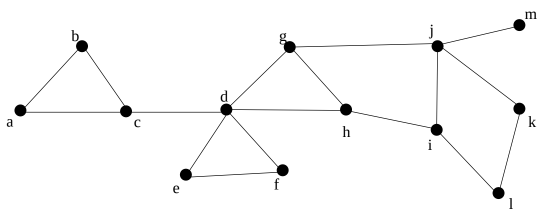
Math 210
Spanning Trees Handout

Definition: Let Γ be a simple graph. A **spanning tree** of Γ is a subgraph of Γ that is a tree containing every vertex of Γ.

Theorem 1: A simple graph is connected if and only if it has a spanning tree
Proof: Next time...

Note: To build a spanning tree for a graph, we often think of removing enough edges to break every cycle with out disconnecting the graph. Spanning trees are useful for building "efficient" networks in a variety of settings.

Algorithms for Finding Spanning Trees:



We will discuss two methods for finding spanning trees, using the graph above as an example.

1. **Depth-First Search**

   In Depth-First Search we carry out the following:

   (a) Arbitrarily choose one vertex to serve as the root of a rooted tree based on the original graph.
   (b) Form a path starting at the root by successively adding vertices, where each vertex is adjacent to the previous vertex in the path, and so that each vertex added is not already in the graph.
   (c) Continue in the fashion until either all vertices in the graph have been added, or until no new vertices can be added to extend the path.
   (d) Back up along the current path until a new vertex can be added by adding a new "branch" out of the path.
   (e) Repeat the previous two steps until every vertex has been added exactly once. The result is a rooted tree that represents a spanning tree of the original graph.

2. **Breadth First Search**

   In Breadth-First Search we carry out the following:

   (a) Arbitrarily choose one vertex to serve as the root of a rooted tree based on the original graph.
   (b) Add in edges leading to every child of the root (the vertices in level 1 of the rooted tree).
   (c) Next, add edges from the level 1 vertices to any adjacent vertices not already in the tree (these vertices form level 2 of the rooted tree).
   (d) Continue in this fashion until all vertices have been added to the tree. The result is a rooted tree that represents a spanning tree of the original graph.

Depth-First Search on our Example: **Breadth-First Search on our Example:**