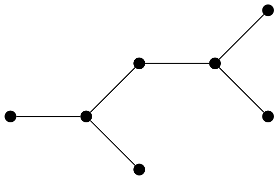
Lecture 32: Trees

A cycle is a closed trail (with at least one edge) that doesn't repeat any vertex except that it ends where it started. A tree is a connected graph with no cycles.

For example,



is a tree.

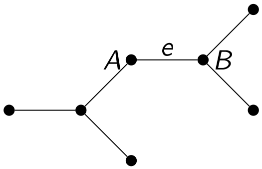
The number of edges in a tree

A tree with n vertices has $n - 1$ edges.

The proof is by strong induction on n.

Base step. A tree with 1 vertex has 0 edges (since any loop would itself be a cycle).

Induction step. Supposing any tree with $j \leqslant k$ vertices has $j - 1$ edges, we have to show that a tree with $k + 1$ vertices has k edges. Well, given a tree T_{k+1} with $k + 1$ vertices, we consider any edge e in T_{k+1}, e.g.



Removing e disconnects the ends A and B of e. (If they were still connected, by some path p, then p and e together would form a cycle in T_{k+1}, contrary to its being a tree.)

Thus $T_{k+1} - \{e\}$ consists of two trees, say T_i and T_j with i and j vertices respectively. We have $i + j = k + 1$ but both $i, j \leqslant k$, so our induction assumption gives

$$T_i \text{ has } i - 1 \text{ edges}, \quad T_j \text{ has } j - 1 \text{ edges}.$$

But then $T_{k+1} = T_i \cup T_j \cup \{e\}$ has $(i - 1) + (j - 1) + 1 = (i + j) - 1 = k$ edges, as required.

Remarks

1. This proof also shows that any edge in a tree is a bridge.
2. Since a tree has one more vertex than edge, it follows that m trees have m more vertices than edges.
3. The theorem also shows that adding any edge to a tree (without adding a vertex) creates a cycle. (Since the graph remains connected, but has too many edges to be a tree.)

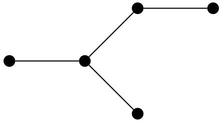
These remarks can be used to come up with several equivalent definitions of tree.

Next we see how any connected graph can be related to trees.

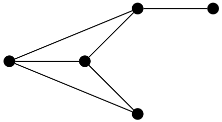
Spanning trees

A *spanning tree* of a graph G is a tree contained in G which includes all vertices of G.

For example,



is a spanning tree of



Any connected graph G contains a spanning tree.

This is proved by induction on the number of edges.

Base step. If G has no edge but is connected then it consists of a single vertex. Hence G itself is a spanning tree of G.

Induction step. Suppose any connected graph with $\leqslant k$ edges has a spanning tree, and we have to find a spanning tree of a connected graph G_{k+1} with $k+1$ edges.
If G_{k+1} has no cycle then G_{k+1} is itself a tree, hence a spanning tree of itself.
If G_{k+1} has a cycle p we can remove any edge e from p and $G_{k+1} - \{e\}$ is connected (because vertices previously connected via e are still connected via the rest of p). Since $G_{k+1} - \{e\}$ has one edge less, it contains a spanning tree T by induction, and T is also a spanning tree of G_{k+1}.

Remark It follows from these two theorems that a graph G with n vertices and $n-2$ edges (or less) is *not* connected.
If it were, G would have a spanning tree T, with the same n vertices. But then T would have $n-1$ edges, which is impossible, since it is more than the number of edges of G.

The greedy algorithm

Given a connected graph with weighted edges, a minimal weight spanning tree T of G may be constructed as follows.

1. Start with T empty.
2. While G has vertices not in T, add to T an edge e_{k+1} of minimal weight among those which do not close a cycle in T, together with the vertices of e_{k+1}.

This is also known as *Prim's algorithm*.

Remarks

1. T is not necessarily a tree at all steps of the algorithm, but it is at the end.
2. For a graph with n vertices, the algorithm runs for $n-1$ steps, because this is the number of edges in a tree with n vertices.
3. The algorithm is called “greedy” because it always takes the cheapest step available, without considering how this affects future steps. For example, an edge of weight 4 may be chosen even though this prevents an edge of length 5 being chosen at the next step.
4. The algorithm always works, though this is *not* obvious, and the proof is not required for this course. (You can find it, e.g. in Chartrand’s *Introductory Graph Theory*.)
5. Another problem which can solved by a “greedy” algorithm is splitting a natural number n into powers of 2. Begin by subtracting the largest such power $2^m \leqslant n$ from n, then repeat the process with $n - 2^m$, etc.

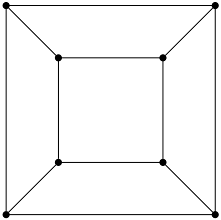
Questions

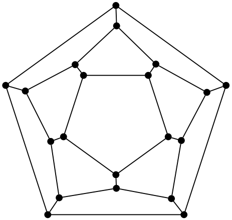
32.1 Which of the following graphs are trees? In each case we insist that $m \neq n$.

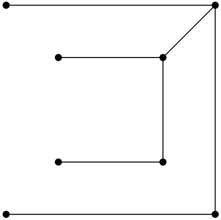
- vertices $1, 2, 3, 5, 7$
  an edge between m and n
  if m divides n or n divides m
- vertices $1, 2, 3, 4, 5$
  an edge between m and n
  if m divides n or n divides m
- vertices $2, 3, 4, 5, 6$
  an edge between m and n
  if m divides n or n divides m

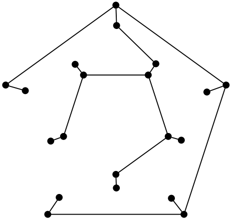
Questions

32.2 Find spanning trees of the following graphs (cube and dodecahedron).





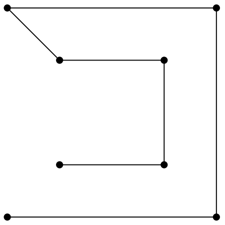




Questions

32.3 Also find spanning trees of the cube and dodecahedron which are paths.

For the cube, here's one answer



I'll leave the dodecahedron as a puzzle for you!