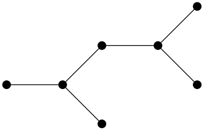
MAT1830

Lecture 32: Trees

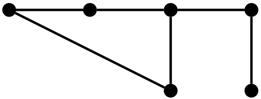
A *tree* is a graph that is connected and has no subgraph that is a cycle.

For example,

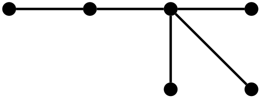


is a tree.

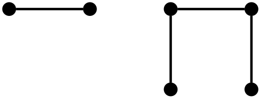
Question Which of the following graphs are trees?



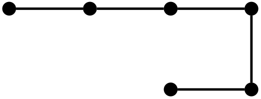
No. Has a cycle.



Yes.



No. Not connected.



Yes. A path is a special kind of tree.

Flux Exercise

Which of the following graphs are trees?
In each of A,B,C put an edge between vertices m and n when $m \neq n$ and either m divides n or n divides m.

A. Graph with vertices 1,2,3,5,7 and edges as described.

B. Graph with vertices 1,2,3,4,5 and edges as described.

C. Graph with vertices 2,3,4,5,6 and edges as described.

D. Both A and C.

E. None of the above.

32.1 The number of edges in a tree

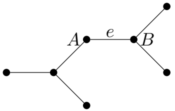
A tree with n vertices has $n-1$ edges.

The proof is by strong induction on n.

Base step. A tree with 1 vertex has 0 edges (an edge requires at least 2 vertices).

Induction step. Supposing any tree with $j \leqslant k$ vertices has $j-1$ edges, we have to deduce 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.

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

Proof Let $P(n)$ be the statement that "each tree with n vertices has $n-1$ edges".

Base step. A tree with 1 vertex has 0 edges, so $P(1)$ is true.

Induction step. Suppose that $P(1), \ldots, P(k)$ are true for some integer $k \geq 1$.

We want to prove that $P(k+1)$ is true: each tree with $k+1$ vertices has k edges

- Let G be a tree with $k+1$ vertices.
- Choose any edge of G and delete it.
- The remaining graph is disconnected but is made of two connected 'pieces', say one has i vertices and the other has j vertices, where $i+j=k+1$.
- Each connected piece has no cycles so is a tree.
- So one piece has $i-1$ edges by $P(i)$ and the other has $j-1$ edges by $P(j)$.
- So the number of edges in G was $(i-1)+(j-1)+1=i+j-1=k$.
- So $P(k)$ is true.

So $P(n)$ is true for each integer $n \geq 1$.

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.

Flux Exercise

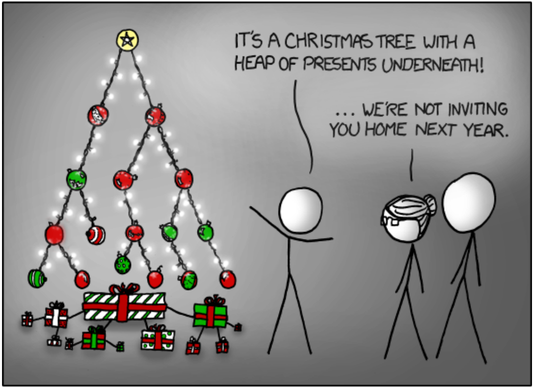
Which of the following might be the sequence of degrees of the vertices of a tree?

A. 1, 1, 1, 1, 1, 2, 2, 3.

B. 1, 1, 1, 1, 2, 2, 3, 3.

C. 1, 1, 1, 2, 2, 2, 3, 3.

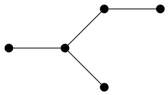
D. 1, 1, 1, 2, 2, 3, 3, 3.


IT'S A CHRISTMAS TREE WITH A HEAP OF PRESENTS UNDERNEATH!
... WE'RE NOT INVITING YOU HOME NEXT YEAR.

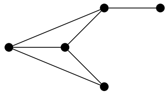
32.2 Spanning trees

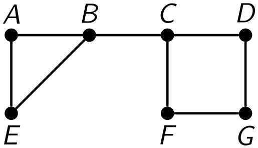
A *spanning tree* of a graph G is a tree that is subgraph of G and includes every vertex of G.

For example,

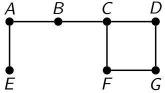


is a spanning tree of

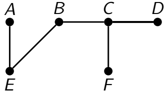




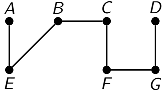
Question Which of the following are spanning trees of the graph above?



No. Not a tree.



No. Not spanning.



Yes.

Question How many spanning trees does the graph have?

Answer 12. Delete any edge in $\{AB, AE, BE\}$ and any edge in $\{CD, CF, DG, FG\}$.

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}.

Fact. Each connected graph has a spanning tree.

Proof Let $P(n)$ be the statement that "each connected graph with n edges has a spanning tree".

Base step. A connected graph with 0 edges has just 1 vertex and is a tree.
So $P(1)$ is true.

Induction step. Suppose that $P(k)$ is true for some integer $k \geq 0$.

We want to prove that $P(k+1)$ is true: each connected graph with $k+1$ edges has a spanning tree.

- Let G be a connected graph with $k+1$ edges.
- If G has no cycle, it is a tree. So it's its own spanning tree.
- Otherwise G has a cycle. Remove any edge of the cycle.
- The remaining graph has k edges and is still connected.
- So the remaining graph has a spanning tree T by $P(k)$.
- But T is a spanning tree for G too.
- So $P(k)$ is true.

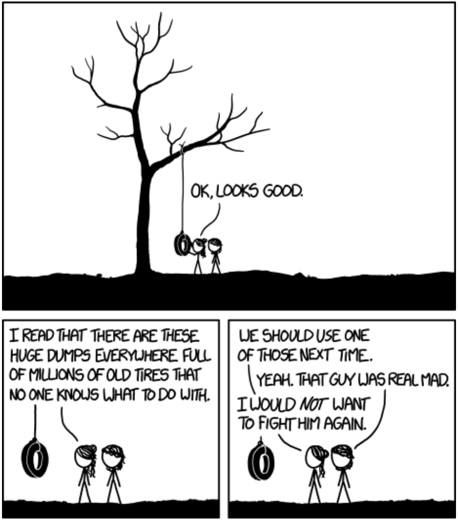
So $P(n)$ is true for each integer $n \geq 0$.

Basic idea "delete edges from cycles until there are no more cycles".

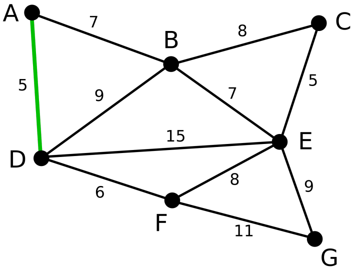
Remark It follows from these two theorems that a graph G with n vertices and $n-2$ edges (or less) is *not* connected.

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.

$\leq n-2$ edges	$n-1$ edges	$\geq n$ edges
cannot be connected	if connected has no cycles	can be connected **and** contain cycles


OK, LOOKS GOOD.
I READ THAT THERE ARE THESE HUGE DUMPS EVERYWHERE FULL OF MILLIONS OF OLD TIRES THAT NO ONE KNOWS WHAT TO DO WITH.
WE SHOULD USE ONE OF THOSE NEXT TIME.
YEAH. THAT GUY WAS REAL MAD.
I WOULD *NOT* WANT TO FIGHT HIM AGAIN.

In many applications it's very useful to start with a graph with **weights** (nonnegative labels) on its edges and find a spanning tree of the graph with a small total weight.

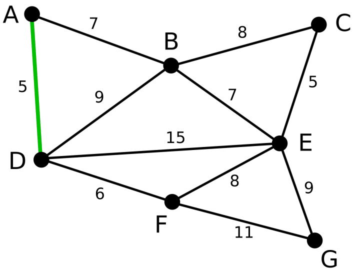


32.3 The greedy algorithm

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 T is not a spanning tree for G, add to T an edge e_{k+1} of minimal weight among those which do not create a cycle in T, together with the vertices of e_{k+1}.

This is also known as *Kruskal'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.

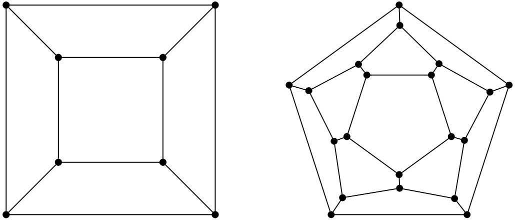
3. The algorithm is called "greedy" because it always takes the cheapest step available, without considering how this affects future steps. For example, an edge of weight 4 may be chosen even though this prevents an edge of length 5 being chosen at the next step.

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*.)

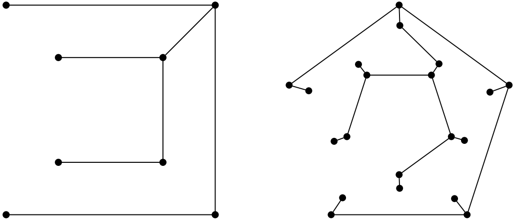
5. Another problem which can solved by a "greedy" algorithm is splitting a natural number n into powers of 2. Begin by subtracting the largest such power $2^m \leqslant n$ from n, then repeat the process with $n - 2^m$, etc.

Question

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



Answer

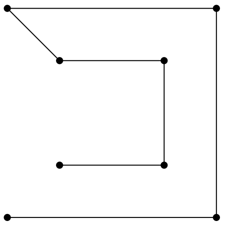


Question

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

Answer

For the cube, here's one answer



Have a go at the dodecahedron yourself.