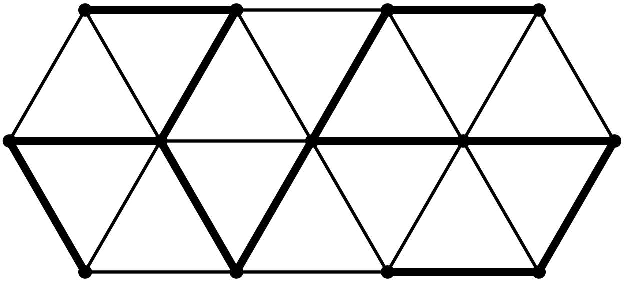
MA2C03 Mathematics
School of Mathematics, Trinity College
Hilary Term 2016
Lecture 33 (January 20, 2016)

David R. Wilkins

33. Spanning Trees

Definition

A *spanning tree* in a graph (V, E) is a subgraph of the graph (V, E) that is a tree which includes every vertex of the graph (V, E).



Theorem 33.1

Every connected graph contains a spanning tree

Proof

Let (V, E) be a connected graph. The collection consisting of all the connected subgraphs of (V, E) with the same vertices as (V, E) is non-empty, since it includes the graph (V, E) itself. Choose a subgraph (V, E') in this collection such that the number $\#(E')$ of edges in this subgraph is less than or equal to the number of edges of any other subgraph in the collection. We claim that (V, E') is the required spanning tree. Clearly (V, E') is connected and has the same vertices as V. It only remains to show that (V, E') does not contain any circuits.

Suppose that (V, E') were to contain a circuit. Let $v\,w$ be an edge traversed by some circuit in (V, E'), and let $E'' = E' \setminus \{v\,w\}$. There would then exist a walk from v to w whose edges belong to E''. (Such a walk could consist of the remaining edges of the circuit traversing the edge $v\,w$.) Moreover every vertex in V could be joined to v by a walk whose edges belong to E', and could therefore be joined either to v or to w by a walk whose edges belong to E''. It would then follow that every vertex of V could be joined to v by a walk whose edges belong to E'', and therefore the graph (V, E'') would be a connected subgraph of (V, E) with the same vertices as (V, E) and with fewer edges than (V, E'), which is impossible. We conclude therefore that the subgraph (V, E') of (V, E) cannot contain any circuits and is therefore the required spanning tree. $\blacksquare$

Corollary 33.1

Let (V, E) be a connected graph with $\#(V)$ vertices and $\#(E)$ edges. Suppose that $\#(E) = \#(V) - 1$. Then the graph (V, E) is a tree.

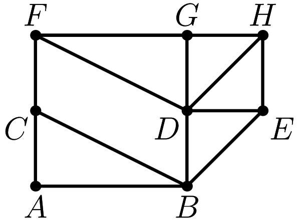
Proof

A connected graph (V, E) contains a spanning tree, by Theorem 33.1. This spanning tree must have $\#(V) - 1$ edges, by Theorem 32.3. But the spanning tree then has the same number of edges as the original graph (V, E), and must therefore be the same as this graph. It follows that the graph (V, E) must be a tree, since it is a spanning tree of itself. $\blacksquare$

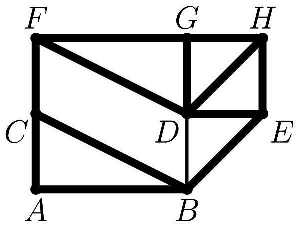
The proof of Theorem 33.1 corresponds to an algorithm for finding a spanning tree for a connected graph. The algorithm proceeds as follows. We start with a subgraph consisting of all the vertices and vertices of the original graph. If that subgraph contains a circuit, then we can remove one of the edges of that circuit. The resultant subgraph will still be a connected subgraph of the original graph that includes all the vertices of the original graph. We can then iteratively break remaining circuits in the subgraph, one by one, so that, at each stage of the algorithm, we have a current subgraph that is connected and includes all the vertices of the original graph. We proceed in this fashion until the current subgraph has no more circuits to break. The subgraph will then be the required spanning tree.

Example

We find a spanning tree for the connected graph with vertices A, B, C, D, E, F, G, H, and edges AB, AC, BC, BD, BE, CF, DE, DF, DG, DH, EH, FG and FH. This graph is pictured below.

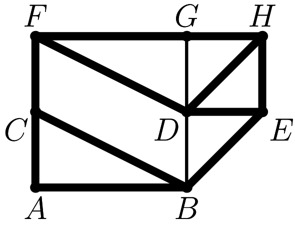


Starting with the current subgraph equal to the given graph, we note that the subgraph has a circuit $B\,C\,F\,D\,B$. We may therefore remove one of the edges of this circuit. Let us therefore remove the edge $B\,D$ from the subgraph. The resultant subgraph is then represented by the thick edges of the diagram below:—

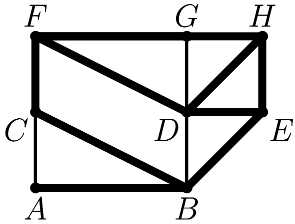


This is the current subgraph for the second removal.

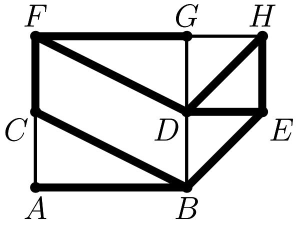
We then break the circuit $D\,F\,G\,D$ of the current subgraph by removing the edge $D\,G$. The resulting subgraph is then the current subgraph for the third removal, and is pictured below.



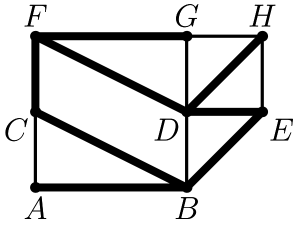
We then break the circuit $A\,B\,C\,A$ of the current subgraph by removing the edge $A\,C$. The resulting subgraph is then the current subgraph for the fourth removal, and is pictured below.



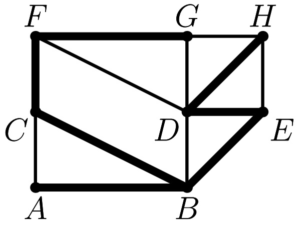
We then break the circuit $D\,F\,G\,H\,D$ of the current subgraph by removing the edge $G\,H$. The resulting subgraph is then the current subgraph for the fifth removal, and is pictured below.



We then break the circuit $B\,C\,F\,D\,H\,E\,B$ of the current subgraph by removing the edge $E\,H$. The resulting subgraph is then the current subgraph for the sixth removal, and is pictured below.



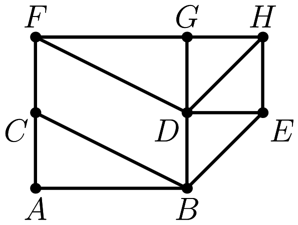
Finally break the circuit $B\,C\,F\,D\,E\,B$ of the current subgraph by removing the edge $D\,F$. The resulting subgraph has no circuits, but is connected and includes all the vertices of the given graph. It is thus a spanning tree for the given graph. This spanning tree is then the subgraph with edges $A\,B$, $B\,C$, $B\,E$, $C\,F$, $D\,E$, $D\,H$, $F\,G$ represented by the thick edges of the following diagram:—



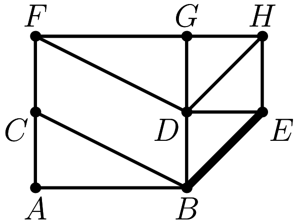
There is an alternative algorithm for finding spanning trees of connected graphs. The procedure is to start with current subgraph of the given graph consisting of just a single vertex. We then add edges one by one, together with any extra vertices incident on those added edges, so as to ensure that, at each stage, the current subgraph is acyclic. When we reach the stage that no further edges can be added to the subgraph without introducing a circuit then the subgraph must be connected and must include all the vertices of the given graph. The subgraph at the final stage must therefore be a spanning tree of the given graph. We illustrate this algorithm by showing how to apply it to find a spanning tree of the graph considered in the previous example.

Example

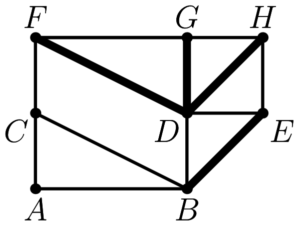
We seek a spanning tree of the graph with vertices A, B, C, D, E, F, G, H, and edges $A\,B$, $A\,C$, $B\,C$, $B\,D$, $B\,E$, $C\,F$, $D\,E$, $D\,F$, $D\,G$, $D\,H$, $E\,H$, $F\,G$ and $F\,H$. This graph is pictured below.



We first add the edge $B\,E$ to obtain the acyclic graph pictured below.

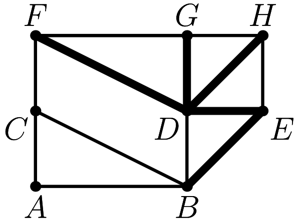


Let us then successively add the edges $D\,F$, $D\,G$ and $D\,H$ (which we can do) to build up the acyclic subgraph, so as to obtain the subgraph pictured below.



It would not then be possible to proceed by adding any of the edges $F\,G$ or $G\,H$ to the acyclic subgraph at the following stage.

We can, for example, add the edge $D\,E$. Adding this edge joins the two components of the acyclic subgraph so as to obtain the acyclic subgraph pictured below.

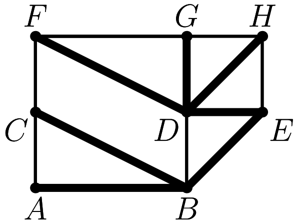


It would not then be possible to proceed by adding any of the edges $B\,D$, $E\,H$, $F\,G$ or $G\,H$ to the acyclic subgraph at the following stage.

The possibilities for the remaining two steps can then be enumerated as follows:—

(i) add AB and then AC;
(ii) add AB and then BC;
(iii) add AB and then CF;
(iv) add AC and then AB;
(v) add AC and then CB;
(vi) add AC and then CF;
(vii) add BC and then AB;
(viii) add BC and then AC;
(ix) add CF and then AB;
(x) add CF and then AC.

For example, opting for possibility (vii) results in the subgraph with vertices A, B, C, D, E, F, G, H and edges $A\,B$, $B\,C$, $B\,E$, $D\,E$, $D\,F$, $D\,G$ and $D\,H$. This subgraph is pictured below.



We now consider the reasons why adding edges to an acyclic subgraph of a given connected graph so as to ensure that the resulting graph remains acyclic is guaranteed to arrive at a spanning tree for the given connected graph.

Suppose that some connected graph is given and that an acyclic subgraph of the given graph has been constructed. Suppose first that the acyclic subgraph does not contain all the vertices of the given graph. Let $\mathbf{v}$ be a vertex of the given connected graph that does not belong to the acyclic subgraph. Then, if any edge incident on the vertex $\mathbf{v}$ is added to the acyclic subgraph, the resulting subgraph will be acyclic, because the addition of an edge incident on the vertex $\mathbf{v}$ cannot result in the formation of a cycle in the resulting larger subgraph.

Next suppose that we have constructed an acyclic subgraph that contains all the vertices of the given connected graph. If this acyclic subgraph is connected then it is a spanning tree. Otherwise there will exist a walk in the given connected graph from a vertex in one connected component of the acyclic subgraph to a vertex in some other connected component. This walk must contain at least one edge whose endpoints are in distinct connected components of the acyclic subgraph. If this edge is added to the acyclic subgraph the resultant subgraph will be acyclic. (The addition of the edge with reduce the number of connected components of the acyclic subgraph by one.)

These observations ensure that if we are given a connected graph, if we have constructed an acyclic subgraph, and if it is impossible to add an edge to that acyclic subgraph so as to ensure that the resulting subgraph is also acyclic, then the acyclic subgraph that has been constructed is a spanning tree for the given connected graph.

This methodology leads to an alternative proof of Theorem 33.1, which asserts that any connected graph has a spanning tree. Indeed any connected graph must contain an acyclic subgraph, where the number of edges in that acyclic subgraph is greater than or equal to the number of edges in any other acyclic subgraph of the given connected graph. It will not then be possible to add an edge to the acyclic subgraph so as to obtain a larger acyclic subgraph. It follows that the acylic subgraph with the maximum possible number of edges must be a spanning tree of the given connected graph.