<u>Corollary</u> Let (V,E) be a connected graph w/ $\#(V)$ vertices and $\#(E)$ edges. If $\#(E) = \#(V) - 1$, Then (V,E) is a tree.

<u>Proof</u> By The previous theorem, every connected graph contains a spanning tree, and by a previous Theorem proven during the section on trees, that tree has $\#(V)-1$ edges $\Rightarrow$ The spanning tree has the same number of edges as (V,E) and is its subgraph by definition $\Rightarrow (V,E)$ is its own spanning tree $\Rightarrow (V,E)$ is a tree.

(q.e.d.)

Constructing spanning trees

<u>Task</u> Given a connected undirected graph, investigate two ways of constructing a spanning tree for it.

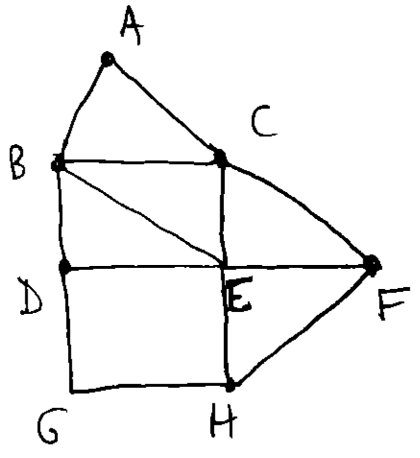
Let (V, E) be a connected undirected graph. We can proceed in one of two ways to construct a spanning tree for it:

(1) Start w/ (V, E) itself. Break up all of its circuits by deleting one edge per circuit.

(2) Start w/ an edge in E. Let this edge be vw. Add back all remaining vertices in $V - \{v, w\}$ by adding in one edge in E per vertex such that at each step the subgraph of (V, E) that we have is both connected <u>AND</u> a tree.

Remark Note that algorithm (1) is akin to the proof of the Theorem that every connected graph has a spanning tree.

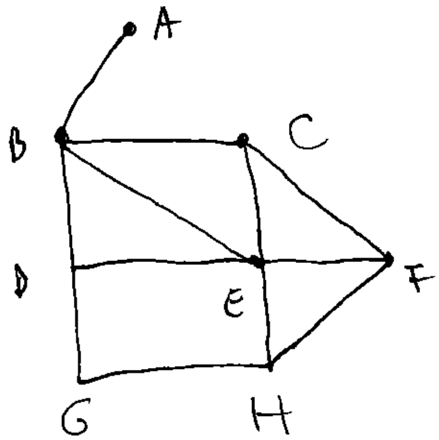
Example Consider



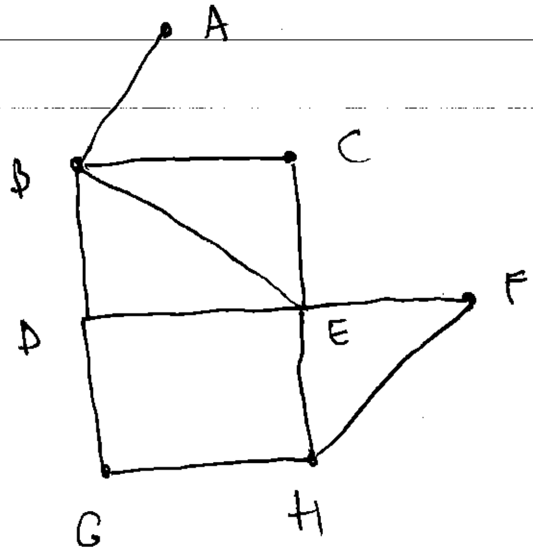
We shall illustrate both (1) and (2) on this graph.

First procedure (1):

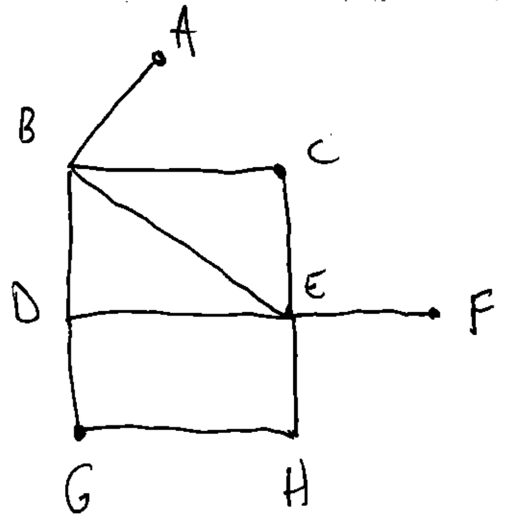
Note ABCA is a circuit. We have a choice which edge to delete. Let us choose to delete AC.



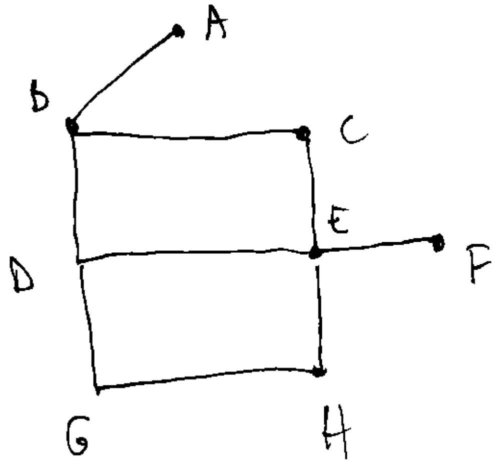
CEFC is a circuit, We choose to delete CF.



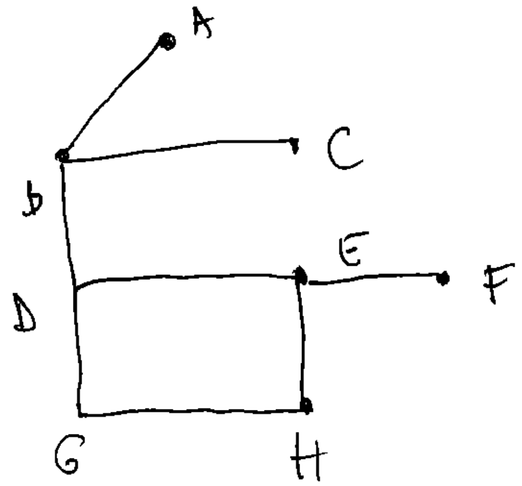
HFHE is a circuit. We choose to delete FH.



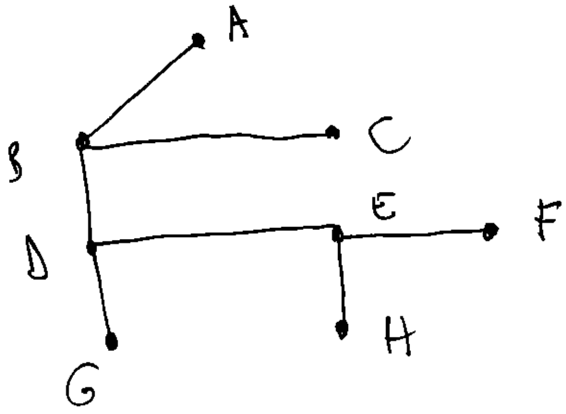
BDEB is a circuit, We choose to delete BE.



BCEDB is a circuit. We choose to delete CE.



DEHGD is a circuit. We choose to delete GH.



The graph that is left doesn't seem to have any circuits. We check that it is a tree using the formula we proved earlier in the course that for a tree $\#(E) = \#(V) - 1$.

$V = \{A, B, C, D, E, F, G, H\} \Rightarrow \#(V) = 8$

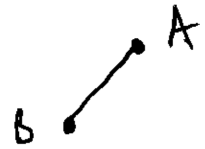
$E' = \{AB, BC, BD, DE, EF, DG, EH\} = 7 = \#(V) - 1$

So (V, E') that we have constructed is a tree and hence the spanning tree of the original (V, E).

Now we follow procedure (2). We start w/ a vertex in V, and at each step we add an edge from E such that this edge is adjacent to a vertex already in the collection of vertices and also to a vertex that is not already in the collection. In other words, at each step, we add a vertex and an edge such that the resulting graph is connected. We stop once we capture all vertices in V.

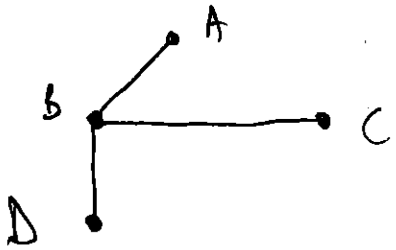
We start w/ vertex A.

We could add vertex B and edge AB OR we could add vertex C and edge AC. We choose to add vertex B and edge AB.

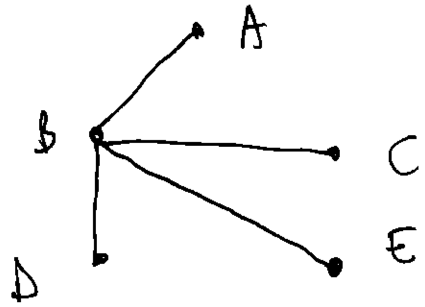


Next, we choose to add vertex C and edge BC

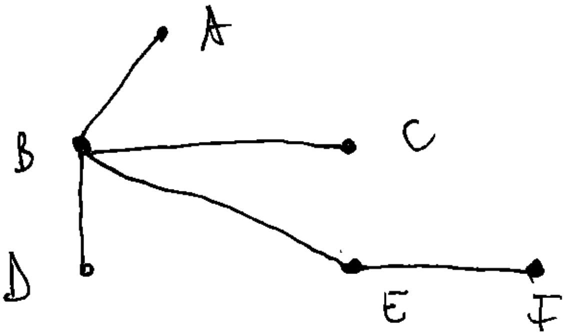
Next, we choose to add vertex D and edge BD



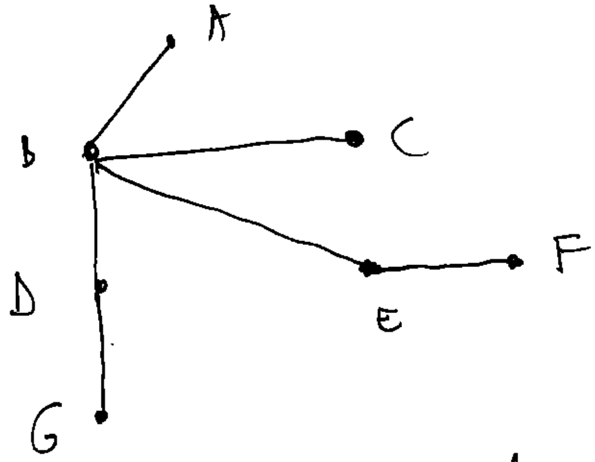
Next, we choose to add vertex E and edge BE.



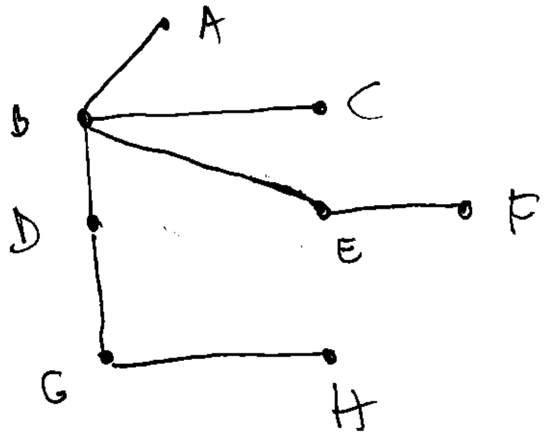
Next, we choose to add vertex F and edge EF.



Next, we choose to add vertex G and edge DG.



Next, we choose to add vertex H and edge GH.



We now have all vertices in $V = \{A, B, C, D, E, F, G, H\}$ 47

We started w/ 1 vertex and 0 edges. At each step we added 1 vertex and 1 edge $\Rightarrow$ at each step i, if V_i is the set of vertices at step i and E_i is the set of edges at step i, we have that $\#(E_i) = \#(V_i) - 1$ for $i = 0, 1, \ldots, 7$.

In other words, at each step, our subgraph (V_i, E_i) is a tree and by construction it is connected. When $V_i = V$, i.e. for $i = 7$, (V, E_7) is a spanning tree of the original (V, E).