Trees: a connected graph without cycles

The following statements about a graph $G$ are equivalent:
- $G$ is a tree
- $G$ is connected with $n$ vertices and $n-1$ edges
- $G$ has $n$ vertices, $n-1$ edges, and no cycles
- $G$ is such that each pair of vertices is connected by a unique elementary chain

Two algorithms for MST problem:

Primes Algorithm
Beginning with a single node, at each iteration a tree is obtained by adding an edge to a node, until all nodes have been included.

Kruskal's Algorithm
Beginning with $N$ trees, each consisting of a single node, at each iteration two trees are combined by adding an edge, until a single tree is obtained.

Example: Prims algorithm for MST

Finding a Minimum Spanning Tree (MST) of a Network (Prim's algorithm)

Step 1 (Setup)
- Select any node to begin the tree

Step 2 (Addition)
- Find a node not currently in the tree which is nearest to the set of nodes in the tree.
- Add that node and the connecting edge to the tree

Step 3 (Stopping criterion)
- If all nodes are in the tree, STOP; otherwise return to step 2
Initially, the tree is empty.
Select (arbitrarily) node A to add to the tree.

Find the node in the set \{B,C,D,E,F\} (not in the tree) which is nearest to the nodes \{A,B\} which are in the tree.
In this case there is a tie!
Break the tie arbitrarily by selecting node C.
Add node C (and edge \{A,C\}) to the tree.

Find the node from the set \{B,D,E,F\} (not in the tree) which is nearest to the nodes (A,B,C) (in the tree).
This is node D, a distance of 5 from the tree.
Add node D (and edge \{A,E\}) to the tree.

Find the node which is nearest to the nodes of the tree (i.e. node A).
This is node G.
Add it (and edge \{A,G\}) to the tree.

Find the node from the set \{B,D,E,F\} (not in the tree) which is nearest to the nodes (A,B,C,G) (in the tree).
This is node E, a distance of 5 from the tree.
Add node E (and edge \{B,E\}) to the tree.

Find the node from the set \{B,D,E,F\} (not in the tree) which is nearest to the nodes (A,B,C,D,E,F) in the tree.
This is node B, a distance of 5 from the tree.
Add node B (and edge \{D,E\}) to the tree.

All nodes are now in the tree, so we stop!
Alaska Gas Transmission Company is planning to construct a pipeline to supply gas from Alaska's north slope (NS) to eight U.S. gas companies, denoted by A through H.

Each mile of right-of-way which is purchased costs an average of $1,000.

How should the pipeline be routed to minimize the total cost of the right-of-way?

Arbitrarily select a node to begin the tree.

Let's choose node NS.

Find the minimum distance from a node NOT in the tree to the node IN the tree.

This is node A.

Add node A (and edge (NSA)) to the tree.

Find the node NOT in the tree which is nearest to the nodes IN the tree.

This is node B, a distance of 12 from node A.

Add node B (by edge (A,B)) to the tree.

Find the node NOT in the tree which is nearest to the nodes IN the tree.

This is node E, a distance of 7 from node B.

Add node E (by edge (B,E)) to the tree.

Find the node NOT in the tree which is nearest to the nodes IN the tree.

This is node D, a distance of 5 from node E.

Add node D (by edge (E,D)) to the tree.

Find the node NOT in the tree which is nearest to the nodes IN the tree.

This is node H, a distance of 12 from node D.

Add node H (by edge (D,H)) to the tree.
A Pl code for Prim's MST algorithm

1.* TGET=BST C1;IN;OUT;KL;DIAG;MIN;J
2. a Compute Minimum Spanning Tree of a graph
3. a
4. IN=1 /* list of nodes in tree
5. OUT=0 /* list of nodes not yet in
6. TREE=C0
7. LENGTH=0
8. a Find shortest arc joining IN & 0UT nodes
9. NEXT=BORDER=1
10. /* IN=1,0UT=1
11. J=BUDDY,J
12. a Add arc from IN node (J) to 0UT node (J)
13. /* IN=1,0UT=1
14. L=OUT1;C2;OUT1;MIN
15. TREE=1;L
16. OUT=OUT+1
17. IN=IN+1
18. LENGTH=LENGTH+1
19. a ENSFIF (OUT)<1

Represent each room, together with the "outside world", by a node, and each gate by an edge.
The problem is to find a spanning tree with the fewest edges.

Kruskal's Algorithm for MST

Step 1: Setup
Let G1=(V1,E1) and i=0
Step 2: Add a new edge
Find (x,y) which minimizes w(x,y), and set w(x,y)= + oo
Step 3: Test for cycle
If the addition of edge (x,y) to the graph G1 would form a cycle, then go to step 2.
Otherwise, add edge (x,y) to graph G1 and increment i.
Step 4: Test for termination
If i < n-1, then return to step 2.
Otherwise, stop with G1 = MST

Example (Kruskal's MST Algorithm)

1. Edges in G1
   0 none
   1 AB
   2 AC, BC
   3 AD, BC, DE
   4 AG, BC, DE, EF

In each of the first 4 iterations, there is a tie for the minimum-length edge to be added.
Example (Kruskal's MST Algorithm)

A network with 15 nodes:

The two trees consisting of nodes 8 & 9 are joined, so that we now have 14 trees:

Edge (6,7) is next added, combining two trees (one with 2 nodes, the other with one), giving us 12 trees:

We begin with 15 "trees", each consisting of a single node:

Next, edge (5,6) is added, which joins two trees, resulting in only 13 trees:

Edge (7,8) is added, combining trees {5,6,7} and {8,9}, giving us only 11 trees:

Since 1+6 = 7, we terminate.
If edge (5, 7) were added, a cycle 5-7-6-5 would be formed, and so we "skip" this edge.

Edge (3, 4) is added next, reducing the number of trees to only 10:

Edge (7, 12) is added, reducing the number of trees to nine:

Edge (14, 15) is added, reducing the number of trees to eight:

Adding edge (10, 11) reduces the number of trees to seven:

Adding edge (5, 9) would create a cycle (5-9-8-7-6-5) and so we don't add this edge.

Adding edge (6, 12) also would create a cycle (6-12-7-6), and so we don't add this edge:

Adding edge (13, 14) doesn't create a cycle, and so we add this edge, reducing the number of trees to only six:
Edge (12, 13) is added, reducing the number of trees to five:

Next we add edge (11, 12), to obtain three trees:

Adding edge (6, 13) would form a cycle, so we skip it:

Adding edge (10, 12) would form a cycle, as would edge (2, 4):

The next edge to be added is (1, 2), which leaves us with only two trees!

Finally, adding edge (2, 6) leaves us with a single tree, spanning all of the nodes!