I didn’t get time to make a very comprehensive handout, so all I have here is a few pointers on Minimum Spanning Trees:

- A **tree** is a graph with no cycles. Think back on the trees you’ve seen earlier in the course to ascertain that this is true.

- Adding an edge to a tree creates a cycle.

- A **spanning tree** is a data structure that is both a tree and also includes all the vertices of the graph as nodes.

- If the edges of the graph have weights, then a tree for which the weights on the edges sum up to the minimum possible number is a **minimum spanning tree**.

- The minimum spanning tree is not unique. Many trees may sum up to the same minimum.

- One of the main algorithms to get a minimum spanning tree is Kruskal’s algorithm.

We’ll do an example of finding a MST using Kruskal’s Algorithm
DFS (which we discussed last time) can be done recursively. Let’s write out and reason about the code for it.
Let’s reason how we can get the longest path that starts from a vertex in the tree without doubling back, using DFS.