

Kruscal's Algorithm

Hello, and welcome back. In this video we will go over the implementation of Kruscal's algorithm.

In most circumstances, at least at the lower to mid level, this is the one you will want to use for computing an MST.

Objectives

Your objective is to implement this algorithm.

The Algorithm

The idea with Kruscal's algorithm is that you select edges in order from lower weight to higher weight and add them to the MST if possible.

To get started, you create a list of edges and either sort them or insert them into a priority queue.

Then, as long as you don't yet have a MST, select edges and add the ones that don't introduce a cycle.

Example

Here's an example. We start by adding a-b and e-g, since they have edge weight of 2. Next if f-g of weight 4. a-c and d-e have weight 4, so we add them. Next we try d-g, also of weight 4, but it would introduce a cycle so we don't add it. Note that we

would have gotten a different MST if we had picked a different order for the edges of weight 4.

b-d has weight 5, and after we add that we are done.

Implementation

...

Here is a sample implementation stolen from the textbook.

We create an edgelist that is a pair; the first part is the weight and the second the pair of edges. Lines 1 to 3 scan the input and add the edges.

Line 5 uses sort to sort the edges. It would be interesting to see if using a priority queue would be faster; if the graph is fairly dense then you might want to try that. It depends also on whether you need to use the edge list for anything else.

Line 6 and 7 initialize some variables. In particular, we use a disjoint set to make sure we don't have any cycles.

From there we loop through all the edges and add the ones that don't make cycles.

Once you are done, you will know how much the MST costs.

Next
Transcript — Minimum Spanning Trees →

