An Alternate Proof to Kruskal’s Algorithm

We give an alternate proof of the correctness of Kruskal’s algorithm for finding minimum spanning trees.

**Claim 1.** Kruskal’s algorithm returns a minimum spanning tree.

*Proof.* We prove it for graphs in which the edge weights are distinct. (Then, to extend it to all graphs requires the usual perturbation argument on the weights that we saw in class.)

Order edges in non-decreasing order of weight, i.e. such that \( w_1 < w_2 < \ldots < w_n \). Suppose by way of contradiction that the spanning tree \( K \) returned by Kruskal’s algorithm is not a minimum spanning tree. Let \( OPT \) denote the minimum spanning tree.

Consider the earliest edge \( e = (a, b) \) on which \( K \) and \( OPT \) disagree. (By earliest edge, we simply mean the edge that is earliest in the order, and by “disagree”, we mean that \( e \) is in one tree and not the other.) It cannot be that \( OPT \) included \( e \) but \( K \) did not, since \( K \) only omits edges if they would create a cycle; if this is the earliest disagreement, then \( e \) in \( OPT \) would be on a cycle and so \( OPT \) could not be a spanning tree. Therefore, the disagreement must be that \( OPT \) did not include \( e \) but \( K \) did.

In \( OPT \), there must be a unique path \( P \) from \( a \) to \( b \); edge \( e = (a, b) \) is not on this path, since \( e \) is not in \( OPT \). Further, there must exist some edge \( e' \) on \( P \) such that \( w_{e'} > w_e \). (Had this not been the case, in other words, had all of the edges of \( P \) been lighter than \( e \), then they would also all be in \( K \), since \( e \) is the first edge on which \( K \) and \( OPT \) disagree. Then \( K \) could not include \( e \) without creating a cycle, a contradiction. Thus, there is some heavier edge \( e' \) on \( P \).)

But now, the tree \( OPT \backslash \{e'\} \cup \{e\} \) is also a spanning tree, and with cost strictly cheaper than that of \( OPT \). This contradicts the optimality of \( OPT \), and thus such a disagreement between \( K \) and \( OPT \) cannot exist.

Note: the identification of two entities \( e \) and \( e' \) to “swap” to get a strictly cheaper optimal solution (and thus a contradiction) is common in the proofs of correctness for several greedy algorithms.