MSTBarbati Logic

First, if the graph has 1 or fewer vertices or if there are no edges, we return an empty list as there are no MSTs, otherwise we start recursion. getMinimumSpanningTrees gets the graph, the list of spanning trees along with a copy of visited, copy of the queue and finally a copy of the current tree.

If the tree's size (plus 1) is equal to the graph's size and it doesn't already exist in the list, we add it to the list and return. While the queue is not empty, we remove an edge from the queue and check if it's source has been visited. If not, create copies of the visited set, queue of edges, and current SpanningTree. Add the edge to the copied tree and add edge's source to the copied visited and incoming edges to the queue and recurse.

Once recursion is done, the list of spanning trees is just that, it contains MSTs but is not only MSTs. If this list has at least one element, find the minimum total weight of the spanning trees in the list, this is the weight of all MSTs. Then we remove all trees that aren't MSTs from the list and return the final list, which contains all MSTs from the graph.