

Process to Find all MSTs

1. Create List of SpanningTrees
2. Find one MST with Prim's algorithm
3. Store weight of that MST
4. Create PQ for edges, HashSet for visited vertices, and SpanningTree to store found MSTs
5. Added 0 vertex to HashSet, and all incoming edges of that vertex to PQ
6. Call getMinimumSpanningTrees recursively
 - a. Compare weight of current tree to weight of MST, if greater, break
 - b. If all vertices have been visited, add tree to List of SpanningTrees
 - c. Otherwise, while the queue is not empty
 - i. Get edge e with least weight from PQ
 - ii. Copy PQ, HashSet, and current SpanningTree
 - iii. Add edge to copied SpanningTree
 - iv. If HashSet does not contain the source vertex of the edge e, add it to the HashSet
 - v. If it is not already in the PQ, add each incoming edge of the source vertex of the edge e
 - vi. Recursively call getMinimumSpanningTrees with copied objects

In summary, the process starts at the 0 vertex and goes from edge with the smallest weight to edge of the smallest edge until all vertices have been visited, like Prim's algorithm. However, at each visited vertex, each incoming edge generates a new recursive call, allowing to algorithm to search depth first through all the edges and generate all the MSTs.