Main Idea

The main idea of our code is to try three different lightweight algorithms and then choose the best result from the three. The three algorithms in our code are SimpleGreedy, ScanGreedy, and Heuristic.

SimpleGreedy

This algorithm is a fairly basic greedy approach that tries to make the tree as wide as possible as fast as possible. The root node is chosen as the vertex with the highest degree, and ties are broken based on lexicographical ordering. All adjacent vertices are then added to the tree as children of the root node. At each iteration, find the leaf with the greatest number of adjacent vertices that are not already in the tree and add them as children of the leaf. Repeat until all vertices are in the tree.

ScanGreedy

This is an improvement to SimpleGreedy in terms of performance, and it only adds another factor of |V| to the runtime. ScanGreedy follows the same algorithm as SimpleGreedy, except it tries all possible root nodes and returns the resulting tree with the greatest number of leaf nodes. It can only do as poorly as SimpleGreedy but offers room for improvement. This room for improvement is how we found "hard" graphs: we could look for instances where ScanGreedy performs much better than SimpleGreedy and/or Heuristic.

Heuristic

This algorithm comes directly from the paper "Approximating Maximum Leaf Spanning Trees in Almost Linear Time" by Hsueh-I Lu and R. Ravi. The worst-case approximation ratio for this algorithm is reported to be 3.

Link to paper: https://www.csie.ntu.edu.tw/~hil/paper/jalg98.pdf