Making Friends Report

Author 1 and Author 2

May 6, 2020

1 Results

All solutions verify as correct, and none take more than a minute. Profiling shows that most time is spent in the find function.

2 Implementation details

The main loop of the function interacts with:

- Remaining: A list of edges (lists of [start node, end node, weight]), sorted by weight. Only the last element is ever removed (constant time operation)
- Tree: A list of weights for edges that have been added. Weights are only ever appended, then sum is called on this list to get the result
- Parent: A list of each node's parent, index in the lit == index of the node. Nodes with no parent have their own index as parent, so we know there's no parent if parent[node index] == node index. This is updated in find and union by changing the value at an index, but no elements are ever added or removed from the list.
- Size: A list of the size of the subtree with a node as it's root. In other words, size[index] gives the number of nodes that have node index as their parent, whether directly or not. This is updated in union by changing the value at an index, but no elements are ever added or removed from the list.