

The graph is stored as a edge list. Each vertex initially acts as its own super-node, and this is tracked using a parent array similar to a Disjoint Set Union . The findParent function tells us which super-node a vertex currently belongs to, so even after several contractions we can always know the current identity of any vertex. The algorithm runs a loop until only two super-nodes remain. In each iteration, a random edge is selected from the edge list. If both endpoints of that edge already belong to the same super-node, the edge is ignored because it has effectively become a self-loop. Otherwise, the two super-nodes are merged by updating the parent array, which contracts that edge. After a contraction, the edge list is rebuilt: every edge is rechecked using findParent, and only edges that connect two different super-nodes are kept. This step removes self-loops automatically .