Worksheet #20: Sequential->Parallel Spanning Tree Algorithm

Insert finish, async, and atomic constructs (pseudocode is fine) to convert the sequential spanning tree algorithm to a parallel algorithm

```
class V {
  V [] neighbors; // adjacency list for input graph
  V parent; // output value of parent in spanning tree
  boolean makeParent(V n) {
    if (parent == null) { parent = n; return true; }
    else return false; // return true if n became parent
  }
  void compute() {
    for (int i=0; i<neighbors.length; i++) {
      final V child = neighbors[i];
      if (child.makeParent(this))
        child.compute(); // recursive call
    }
  }
}

root.parent = root; // Use self-cycle to identify root
root.compute();
...}
```