In this assignment, you will develop data structures and associated code to read in an undirected graph G (given in edge list format), set up a linked adjacency structure for G, and carry out the graph operation of "coalescing" two distinct nodes. Specifically, to coalesce nodes v and w into a new node x, we replace the two nodes by a new node x in which node x is adjacent to any node that either v or w (or both) was adjacent to in the original graph G = (N, E). The resulting graph G_{vw} should have no loops or multiple edges.

Input Data: Assume that the node set $N \subseteq \{1, 2, ..., n\}$, with *n* denoting the largest node number appearing. The graphinput file will have as its first record *n*, and succeeding records will give *G* in edge list form {(FROM, TO)}, with no particular order to the edges. The mergenodes file will contain pairs of nodes *v* and *w* to be coalesced. For simplicity, let the node resulting from merging *v* and *w* be the numerically smaller of {*v*, *w*}. This file can contain several node pairs, so the coalescing operation should be done sequentially in the order given.

1. Using your language of choice, write a procedure SETUP to read graphinput and set up an appropriate linked adjacency structure for G. Do not use special functions built into the language.

2. Develop any auxiliary data structures to efficiently carry out the coalescing operation COALESCE. Your code should also write out, after the sequence of coalesce operations, the final graph G' — the current set of nodes and a list of the current edges in edge list form.

3. Analyze the time complexity of carrying out your SETUP and COALESCE operations in terms of n = |N| and m = |E|.

Turn in adequately documented code for this problem. Accompany this with a plain English description of your overall approach, the data structures used, and important program variables. Discuss how you handled any particularly troublesome design issues.

Suggestions: Try to avoid lots of special cases. Make your code modular, so that pieces can be reused later.

Provide appropriate output for the test data set supplied.