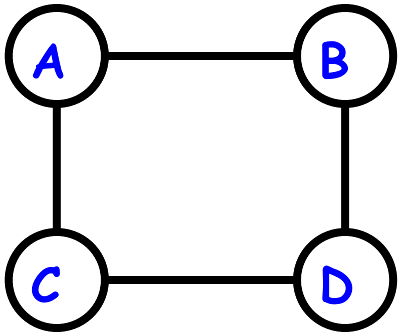
Worksheet #20: Identifying conflicts in isolated constructs

Name: ______________________ Netid: ______________________

Consider the Parallel Spanning Tree algorithm discussed in the last lecture (and shown below in slide 18). Assume that the isolated construct is implemented using a Transactional Memory mechanism. Outline a parallel execution scenario for the input graph below that could lead to a conflict between isolated constructs.



Parallel Spanning Tree Algorithm

```
class V  {
  V [] neighbors; // adjacency list for input graph
  V parent; // output value of parent in spanning tree
  boolean tryLabeling(final V n) {
    return isolatedWithReturn(() -> {
             if (parent == null) parent = n;
             return parent == n; // return true if n became parent
           });
  } // tryLabeling
  void compute() {
    for (int i=0; i<neighbors.length; i++) {
      final V child = neighbors[i];
      if (child.tryLabeling(this))
        async(() -> { child.compute(); }); // escaping async
     }
  } // compute
} // class V
. . .
root.parent = root; // Use self-cycle to identify root
finish(() -> { root.compute(); });
. . .
```