Problem 1 (Problem 22.2-7)

Solution:

1) main idea: we can build up a graph G=(V, E). The vertices are the wrestlers and the edges are the rivalries. Then we run the BFS from the root and save the path-length from each vertex to root. All vertices with odd length are the bad guys; all vertices with even length are the good guys. Then we examine every edge. If the edge is between two vertices whose path lengths to root are both even or both odd, then we cannot establish a rivalry between two Good or Bad guys. otherwise, we return true – it is possible to partition the wrestlers. Therefore, we have |V|=n, and |E|=r

2) pseudocode:

```
Input: graph G (V,E)
Output: True or False
Q = []
Q. enqueue(root)
While Q is not empty:
        p = Q. dequeue ()
        for n is adjacent vertices of p do
                if n is unvisited
                        Q. enqueue(n);
                        mark n as visited
                        save length from r to n as n.length
                End
        End
End
For e (u,v) in E:
        If  (u.length and v.length are odd) or (u.length and v.length are even):
                Return False
Return True
```

---

3) proof: by running BFS, we iterate over all the vertices in the graph and assign the odd or even length to these vertices. It is clear that if there exist this graph, all edges should

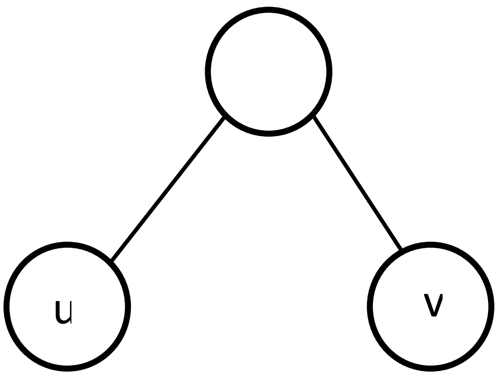
represent a rivalry between two nodes with even and odd length from root. Otherwise, this s impossible.

4) time: O(V+E)

Problem 2 (Problem 17.2-3)

Solution:

If we do depth search of following graph:



When running DFS, it goes from root → u → root → v → root. In this case, u.d < v.d but u and d are siblings.