

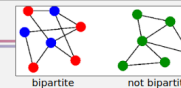
HW5

- be careful to always consider vertices in alphabetical order (when stated)

HW5

- #2 – bipartite graph
 - intent was to trace the 2-coloring / bipartite graph detection algorithm from class rather than simply inspect the graph / look for odd-length cycles

BFS-Based Algorithms



- bipartite graph detection / two-coloring

- a *bipartite* graph is one whose vertices can be divided into two sets such that every edge connects a vertex in one set with a vertex in the other
- *coloring* refers to assigning labels (colors) to vertices so that no two adjacent vertices have the same label (color)
 - a two-coloring uses two colors

```

color[s] = 0
run bfs(s), setting color[v] = the opposite color of
color[u] for each discovery edge (u,v) and checking that
color[v] is the opposite color of color[u] for each non-
discovery edge (u,v)
    - if there is an edge (u,v) for which color[u] = color[v], the graph
      is not bipartite / two-colorable
    
```

- Intuition – following a path along discovery edges must alternate colors, since those edges are graph edges
- can't change the color of any vertex without changing them all
 - non-discovery edges are also graph edges, and ends must be opposite colors

the main error with tracing the algorithm was not detecting a problematic coloring earlier enough

HW5

- #3 – DFS
 - intent was to trace the recursive DFS from class

```

dfs(G,s)
for each vertex u in V-{s} do
    state[u] = "undiscovered"
    prev[u] = null
state[s] = "discovered"
prev[s] = null
dfsHelper(G,s)

dfsHelper(G,u)
process vertex u (early)
for each edge (u,v) in G.incidentEdges(u) do
    if state[v] = "undiscovered" then
        process edge (u,v)
        state[v] = "discovered"
        prev[v] = u
        dfsHelper(G,v)
state[u] = "processed"
process vertex u (late)
    
```

each vertex is discovered right before recursing on it – the neighbors are not all discovered at once

HW5

- #4 – cut vertices
 - the problem asked to find the cut vertices – so be sure to identify the cut vertices!
 - drawing the DFS tree with forward and back edges + entry times makes determining the earliest reachable ancestor and the cut vertices much easier
 - children and descendants are much easier to identify

