

# CSCI 3104 Assignment 7

10:00 - 10:50 Wanshan

Jackson Chen

April 4 2016

1. Assign a color to the source vertex. The color all the neighbors the opposite color. Do the same for the neighbors of neighbors until the entire tree is traversed. While traversing the tree, if a neighbor is already colored with the same color as the current vertex, then it means that the graph cannot be a bipartite graph.

2. Look at back of page for a hand-drawn graph.

3. General idea for a DFS:

Start the root node. Neighbors are defined by R values that connect two jobs. Traverse the tree.

Whichever traversal encompasses every node is a working job schedule. Here is the pseudocode:

```
nodes = [j1, j2, ..., jn]
r = [r1, r2, ..., rn]
for (int i = 0; i < n; i++) {
    nodes[i].neighbor.append(r[i])
    nodes[i].color = "white"
    node[i].pi = nil
    // Creates tree
}
time = 0
for (int i = 0; i < n; i++) {
    if (nodes[i].color == "white"):
        lastVisited = DFSvisit(G, nodes[i]) // Visits node
}
// Check if all nodes visited
all = true;
for (int i = 0; i < n; i++) {
    if (nodes[i] != "black"):
        all = false;
}
if (all):
    print lastVisited
    tmp = lastVisited
    while (tmp.pi != nil):
        print tmp.pi
        tmp = tmp.pi
```