CS62 - Connected Components

David Kauchak

Note that the code below is pseudocode: code that is used to describe an algorithm, but is not give the full, compilable code.

• Some general things to think about/talk about:
  – What does the method do? Explain what the role of the different parameters is, what is returned and how the method operates.
  – Show some examples, particularly with multiple connected components.
  – What is the running time of the method with respect to $|V|$ the number of vertices and $|E|$ the number of edges?
    * How many times is each vertex visited?
    * How many times is each edge visited/examined?

• Some specific things to think about/talk about:

  – What exactly is returned by the method?
  – What does the for loop in `grop_connected_components` do? Why do we need it?
  – `current` is a local variable. Why can we add it to the `connected_components` list?
void dfs_get_component(int v, list<int> &component) {
    set v as visited
    add v to component

    for (all neighbors u of v) {
        if (we have NOT visited u ){
            dfs_get_component(u, component);
        }
    }
}

list<list<int> > grop_connected_components(const map<int, list<int> >& adjMap){
    list<list<int> > connected_components;

    for (all vertices v){
        if (we have NOT visited v) {
            list<int> current;
            dfs_get_component(v, current);
            connected_components.push_back(current);
        }
    }
}