CSCI 2270

Data Structures & Algorithms

Gabe Johnson

Lecture 31 Apr 5, 2013

Graph Coding, Part 2

Upcoming Homework Assignment

HW #8 **Due: Mon, Apr 8**

Graph DFS and BFS

The DFS/BFS homework is due on Monday. Going to code all day today (again) and I'll upload whatever I do in class. About 1/3 of you have gotten started.

```
0: 8     | ********
1: 0     |
2: 0     |
3: 0     |
4: 6     | ******
5: 0     |
6: 9     | *********
7: 0     |
8: 0     |
9: 2     | **
10: 0    |
11: 7    | *******
12: 0    |
13: 0    |
14: 0    |
15: 10   | **********
Total students participating: 42
```

Lecture Goals

1. Node Color
2. Discovery Times
3. ‘The other end of the edge’
4. Edge Types

Node Color

Rather than

```
nd->color = BLACK;
```

You need to use the setColor member func:

```
nd->setColor(BLACK, clock);
```

Disco Time + Clock

The discovery (and finish) time for a node should be unique. And, every time step should be accounted for. You should increment the clock *immediately before* changing a node's color to GRAY or BLACK.

```
clock = clock + 1;
nd->setColor(GRAY, clock);
```

Other end of edge

Say we have a Node* **n** and an Edge* **e**. We know that **e** is adjacent to **n** but we don't know which end of **e** it is. It could be the start *or* the end node of the edge.

```
Node* n = some node.
Edge* e = some edge adjacent to n.
Node* other = Node in e that isn't n.
```

What's the code to do that last step?

Other end of edge

```
Node* n = some node.
Edge* e = some edge adjacent to n.
Node* other; // declare, don't assign yet
if (e->getStart() == n) {
  other = e->getEnd();
} else {
  other = e->getStart();
}
```

Pro tip: turn this into a helper func.

This works because `n, e->getEnd()` and `e->getStart()` are all pointers. Comparing their *values* compares the address of the Node objects.

Coding Session

We'll spend the rest of the day coding up the `Graph::dfs(Node)` function.