Homework 8 due 10 pm today, Tuesday November 22nd

Program 4 due this Sunday 10 pm Sunday, November 27th
Consulting available until 4pm on Wed

Last Time
Red-Black Trees
- Insert Operation
  - Must preserve red-black properties
  - If K and P are red and P’s S is null, or black – TNR
  - If K and P are red and P’s S is red, -- RC
  - May need to cascade changes up the tree to the root

Today
ADTs/Data Structures
Graphs
- terminology
- implementation
- edge representations

Next Time
Graphs
- traversals
- applications
-
Graph Terminology
Implementing Graphs

Graph ADT Ops

Graph Class

Graphnode Class
Representing Edges

Adjacency Matrix

Given the following graphs:

Graph 1

Graph 2

Show the adjacency matrix representation of the edges for each of the graphs:

Graph 1

Graph 2
Representing Edges

Adjacency Lists

Given the following graphs:

Graph 1

Graph 2

→ Show an adjacency list representation of the edges for each of the graphs:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph 1</td>
<td>Graph 2</td>
</tr>
<tr>
<td>0:</td>
<td>A:</td>
</tr>
<tr>
<td>1:</td>
<td>B:</td>
</tr>
<tr>
<td>2:</td>
<td>C:</td>
</tr>
<tr>
<td>3:</td>
<td>D:</td>
</tr>
<tr>
<td>4:</td>
<td>E:</td>
</tr>
</tbody>
</table>
Using Edge Representations

→ Write the code to be added to a Graph class that computes the degree of a given node in an undirected graph.

1. Adjacency list:

   public int degree(Graphnode<T> n) {

2. Adjacency matrix:

   public int degree(Graphnode<T> n) {


Comparison of Edge Representations

Ease of Implementation

Space (memory)

AM

AL

Time (complexity of ops)

node’s degree?

AM

AL

edge exit between two given nodes?

AM

AL