CS 367 - Introduction to Data Structures  
Week 11, Spring 2017

Midterm Exam 2
- TUESDAY 4/4 5:00 pm – 7:00 pm
- Lecture 1: Room 272 of Bascom Hall
- Lecture 2: Room B10 of Ingraham Hall
- Lecture 3: Room 2241 Chamberlin Hall
- UW ID REQUIRED
- Bring #2 pencils
- See posted exam information

Program 4 due 10 pm Sunday, April 16th

Last Week
Classifying Binary Trees, Balanced Search Trees, Red-Black Trees: tree properties, print, lookup, insert

This Week
Read: Red-Black Trees, Graphs
Exam Review (bring Exam 2 Sample Questions)

Finish Red-Black Trees (bring notes from week 10)
- Insert

ADTs/Data Structures Revisited

Graphs
- terminology
- implementation
- edge representations

Next Week (more Graphs)
Read: continue Graphs
Graph Terminology
Implementing Graphs

Graph ADT Ops

Graph Class

Graphnode Class
Representing Edges

Adjacency Matrix

Given the following graphs:

Graph 1

0 → 1
4 → 2
3 → 2

Graph 2

A → B
E → C
D → C

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

Graph 1

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Graph 2

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Representing Edges

Adjacency Lists

Given the following graphs:

Graph 1

0: 1
1: 0
2: 3
3: 4
4: 2

Graph 2

A: B
B: A
C: E
D: C
E: D

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

```java
public int degree(Graphnode<T> n) {
```

2. Adjacency matrix:

```java
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
ADTs/Data Structures

Linear

- predecessors: at most 1
- successors: at most 1

Hierarchical

- predecessors: at most 1
- successors: 0 or more - general tree, at most two - binary tree

Graphical

- predecessors:
- successors: