FINAL EXAMINATION – MAY 2006
CSC 112
NICHOLAS R. HOWE

This is a closed-book exam. You may use two double-sided 8.5x11 sheets of notes.

All answers to this exam should be written in your exam booklet(s). Start with the questions that you know how to do, and try not to spend too long on any one question. Partial credit will be granted where appropriate. You will have two hours and twenty minutes. Good luck!

Data Structures

1. (10 points) Select the most appropriate data structure for the following problems. Be as specific as possible; if you can supply a more specific name than a general data structure ("binary tree" vs. "tree"), then you should do so.

a. You are writing a program that will direct victims of natural disasters quickly to the service agency most able to help them. The program will ask a series of questions, beginning with general ones and becoming more specific as it learns more about each case.

Decision tree

b. You are writing code for a new browser, and want to implement "cookies". These are pieces of data identified by a name (character string); when a web site presents the name the browser must be able to quickly return the associated data.

Hash table

c. You are writing software to calibrate a scientific instrument. The key measurement is dependent on the current temperature. You wish to be able to take the current temperature and quickly look up the proper correction to the actual measurement. (You already know the corrections for every tenth of a degree in the operating range of your device.)

Array

d. You wish to model trade between the nations of the world. For example, you might want to identify groups of countries that trade mostly amongst themselves.

Graph

e. You are writing a program to sequence segments of DNA. You will begin with short fragments, and assemble them into longer ones by stitching together two adjacent sequences.

List

Trees

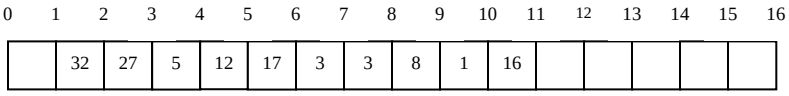
2. (18 points) Write a few paragraphs defining and differentiating the following: arrays, lists, binary search trees, red-black trees, B-trees, hash tables. You should use the differences in element lookup and insertion on the different data structures as an organizing standard of

comparison. In the process, explain the reason for any distinctive features of these data structures.

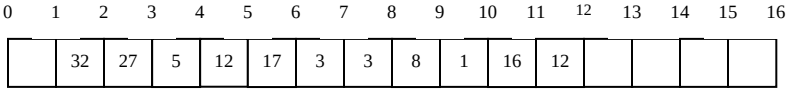
Arrays and lists both store data, but arrays are slow for insertion/deletion and lists are slow for lookup/access. Binary search trees are as fast as lists for insertion deletion but faster for lookup. However, they can become unbalanced, in which case they devolve into lists. Red/black trees are designed to always stay in balance by applying rotation operations as necessary. B-trees are also balanced trees, but with as many branches per node as possible so as to limit disk access. Hash tables have faster access than any of trees, but it is not possible to generate a sorted list quickly.

Heaps

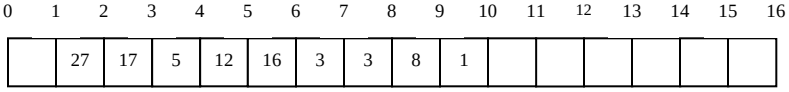
3. (16 points) Suppose that the array below implements a heap in the manner described in class, where level k of the tree is stored in the array at consecutive indices from 2^{k-1} to 2^k-1. Draw the condition of the heap after the following operations are complete. (The operations are not cumulative; for each part you should assume that the heap begins in the configuration shown.)



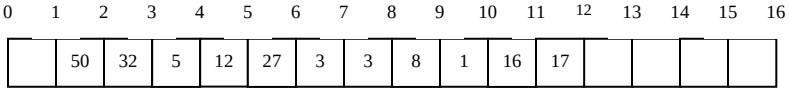
a. Insert 12.



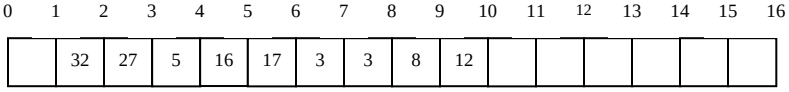
b. Remove 32.



c. Insert 50.

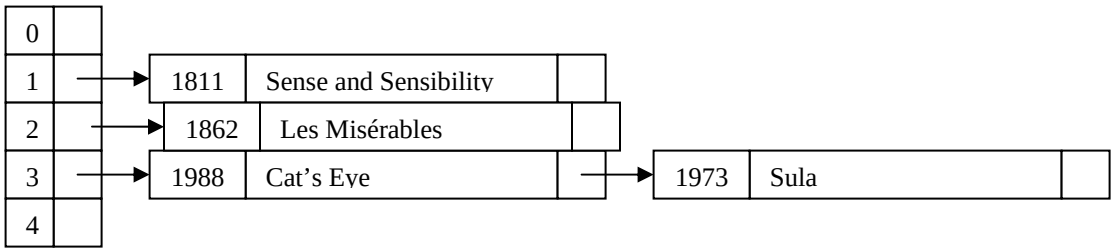


d. Remove 1.



Hash Tables

4. (15 points) The following operations are to be performed in the hash table below, which contains just five slots. The table uses chaining (i.e., linked lists) in case of collisions. Draw the state of the table showing the cumulative effect of all the indicated operations.



a. Insert (1862, Les Misérables).

b. Insert (1973, Sula).

c. Lookup (1353).

d. Remove (1353).

e. Insert (1988, Cat's Eye).

Recursion

5. (16 points) Simulate the execution of the following program. What would be its output? (You may find it helpful to draw a tree to keep track of the recursion.)

```
public class Ack {

    public static int ack(int m, int n) {
       System.out.println("("+m+","+n+")");
       if (m == 0) {
           return n+1;
       } else if (n == 0) {
           return ack(m-1,n);
       } else {
           return ack(m-1,ack(m,n-1));
       }
    }

    public static void main(String[] args) {
       ack(2,1);
    }
}
```

(2,1)
(2,0)
(1,0)
(0,0)
(1,1)
(1,0)
(0,0)
(0,1)

Program Simulation

6. (10 points) Simulate the execution of the following program. What would be its output? (Keep in mind the differences between primitive and reference types as you do so.)

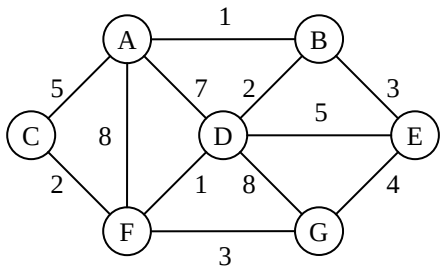
```
public class Sim {
  static int x = 1;
  static int y = 2;

  public static void main(String[] args) {
     int x = 3;
     int y = 4;
     Data d = new Data();
     d.method1(x);
     System.out.println(x);
     System.out.println(y);
     System.out.println(d.x);
     System.out.println(d.y);
     int[] z = new int[1];
     z[0] = Sim.x;
     d.method2(z);
     System.out.println(Sim.x);
     System.out.println(Sim.y);
     d.method3(d);
     System.out.println(x);
     System.out.println(y);
     System.out.println(d.x);
     System.out.println(d.y);
  }
}
```

3
4
5
8
9
11
3
4
12
14

```
public class Data {
  int x;
  int y;

  public Data() {
    x = 5;
    y = 6;
  }

  public void method1(int x) {
    x = 7;
    y = 8;
  }

  public void method2(int[] z) {
    Sim.x = 9;
    z[0] = 10;
    Sim.y = 11;
  }

  public void method3(Data d) {
    this.x = 12;
    y = 13;
    d.y = 14;
  }
```

Graphs

7. (15 points) Consider the undirected graph at right when answering the following questions. Whenever multiple nodes are eligible to be chosen next, assume that they are picked in alphabetical order.



a. In what order would the nodes first be visited when doing a breadth-first traversal starting from C?

C,A,F,B,D,G,E

b. In what order would the nodes first be visited when doing a depth-first traversal starting from G?

G,D,A,B,E,C,F

c. In what order would the nodes be visited by Dijkstra's shortest-path algorithm starting from A?

A,B,D,E,F,C,G