Csc 115 Nov 21st, Fall 2002

Csc 115 – Midterm 3 hints

Midterm III Tips

- Readings:
 - Chapters 8.1-8.3 and 7 (except 7.4)
 - Review Chapters 1 and 2 (OO programming)
- Notes:
 - Heaps and priority queues
 - Pair programming
 - Dictionaries and hashing
 - Model View Controllers and Introduction to Swing
 - Recursion revisited and some analysis of algorithms (see chap 3)
- Code examples (look at all discussed since last midterm):
 - BinaryHeap.java HashEntry.java HashTable.java
 - RecursiveTree.java TreeNode.java
 - Button.java Button2.java Button3.java
 - BTree.java TreeDemo.java
 - DynamicTree.java DynamicTreeDemo.java
 - Stars.java Mystery.java Factorial.java
 - Reverse.java Permute.java Tower.java

Csc 115 – Midterm 3 hints

Midterm Review – Heaps and Queues

- 1. What are the basic operations in a priority queue?
- 2. Operations such as search() and member() are not supported why?
- 3. Write the pseudocode to show how you would implement a sorting algorithm using a priority queue
- 4. What is the running time of this algorithm? Justify your answer
- 5. Given the following array which is a heap encoding of a priority queue, draw the corresponding tree (assumption complete binary tree)

Midterm Review – Heaps and Queues 2

- 6. What is meant by the "shape property" and "order property" of a valid heap encoding?
- 7. For the following tree, show what happens when a deleteMin() operation is called (draw trees to show the intermediate steps of this operation:



- 8. What is the complexity of a delete operation? Justify your answer.
- 9. Repeat 7/8 for insert min.
- 10. How would you do a sort using heap (heap sort), complexity?

Csc 115 – Midterm 3 hints

Midterm Review – Pair programming 3

1. Write a paragraph to describe what is meant by "pair programming". In particular describe how it differs from the general approach of dividing and conquer which is common in computer science courses.

Midterm Review -- Hashing 4

- 1. Why are computer dictionaries more powerful than paper dictionaries?
- 2. What are the basic operations that you would need to provide for a simple dictionary?
- 3. As a reminder here is a list of some of the methods specified by the java.util.Map class. If the get method returns null, does this mean that the key does or does not exist in the map? Is there any way to be sure about our answer?

```
boolean containsKey(Object key)
boolean containsValue(Object value)
Object get(Object key)
boolean isEmpty()
Object remove(Object key)
int size()
```

Midterm Review – Hashing 5

- 4. Dictionaries can be either implemented as log files or as hashtables. Which applications would suit a log file implementation? Justify your answer!
- 5. A hashtable is made up of two major parts, what are they? Why is a hashtable sometimes preferable over a log file?
- 6. When we are mapping keys to indices in the bucket array, it is important to have a good hash function, why? If the load factor is above .6 what should we do?
- 7. For keys that are strings, and an array of size N, design a hash function that will map the keys to elements in the array. Is your approach sensitive to the location of the letters in your string?
- 8. When we use a hashtable, sometimes we will have collisions when we map our keys to the bucket array – describe two ways for managing collisions (note: an alternative way to ask this question will be to show an array and ask you to map elements to an array using a particular collision scheme – see the applet posted on the site).

Midterm Review -- Model View Controller 6

- 1. What are the advantages of using the Model View Controller design pattern in your programs? Describe an example application that would benefit from the use of this pattern and say why.
- 2. Look carefully at examples posted on the web something similar will be on the exam and you will have to either document the code or add some code to create a new component or change the display of a component. You will be provided with any necessary Java APIs – however, get practice reviewing them and looking up components, methods etc.

Midterm Review – Recursion 7

- 1. Given a string, how would you generate all possible permutations of it? Show the resulting code (or I may just ask for pseudocode).
- 2. For the Swedish pancake problem, describe how you would do it recursively.
- 3. For some "mystery" code, analyze if it correct and if it could possibly go into an endless loop