Fall 2014

Midterm 2 Review Worksheet

Recitation 11

1. Is the following code good or will it cause a compiler error/a run-time error. How can we fix it, if necessary?

```
ArrayList<Integer> my1 = new ArrayList<Integer>( );
ArrayList<KeyedItem> my2 = new ArrayList<KeyedItem>( );
my1.add("42");\input{Midterm2Problems.tex}

if (my1.contains(42)) {System.out.println("The Answer!");
my2.add(new KeyedItem(12, "December"));
my2.add(new KeyedItem("November", 11));
int x = my2.indexOf(12);
if(my2.get(0).getKey( ).compareTo(my2.get(1).getKey( )) < 0)
     {
     System.out.println("in order");
     }                                                 // change add above
ArrayList<Object> my3 = my1;                           // change Object to Integer
```

2. Using Rosen's definition of height, give an example and a brief explanation of:

   (a) A full m-ary tree of degree 2 and height 2

   (b) A balanced, but not complete tree of degree 3

   (c) An unbalanced binary tree

3. A full 4-ary tree with 256 leaves has ______ internal nodes and ________ vertices

4. True or False:

(a) A tree has only one simple path between any two of its nodes

(b) An ordered rooted tree is uniquely determined when a list of vertices generated by a preorder traversal of the tree and the number of children of each vertex are specified

(c) A rooted tree is called an m-ary tree if every internal vertex has no more than m children

5. Name four cases for deleting a key from a binary search tree

6. construct a binary search tree for the following values: {7, 5, 8, 2, 12, 6, 4, 10, 8, 24}

7. construct a Max-Heap for the following values: {15, 21, 83, 71, 54, 60, 55, 88, 29, 72}

8. If the following is an array implementation of a Max-Heap, delete one element from it, and show the resulting array: {43, 41, 40, 30, 33, 14, 19, 3, 26, 22}

9. True or False

   (a) A heap is a special case of a binary search tree

   (b) A priority queue can be implemented using a binary search tree

   (c) Elements can be deleted from arbitrary positions within a priority queue

   (d) To rebuild a heap after removing the root, take the leftmost right node and put it into the root position

   (e) When using an ArrayList implementation of a heap, insertion is $O(log\, n)$ worst case

   (f) A heap and a binary search tree have the same basic operations: insert, delete, and retrieve

(g) In an ArrayList implementation of a heap, the left child of position x is at position 2x+1

(h) Like MergeSort, HeapSort requires additional memory

Generics:

1. What is a generic?

2. How do you declare a generic?

3. What is the advantage of using generics?

4. What classes have you used that are generics?

Iterators:

1. What is an iterator?

2. How do you declare an iterator?

3. Why might you want to use an iterator?

4. Are languages with iterators more powerful than other languages?

Priority Queues:

1. What is a Priority Queue?

2. How is a Priority Queue different from a queue?

3. How is a Priority Queue different from a stack?

4. Why might you want to use a Priority Queue?

5. What is the complexity of removing an item?

6. What is the complexity of adding an item?

Heaps:

1. What is a Heap?

2. Name and give a brief description of two properties necessary to satisfy a heap:

3. Why might you want to use a heap?

4. What is the complexity of removing an item?

5. What is the complexity of adding an item?

6. What is heapsort?

Trees:

1. What is a Tree?

2. What is a binary tree?

3. What is a leaf?

4. What is a root?

5. What is an interior node?

6. What is a full tree?

7. What is a complete tree?

8. What is a balanced tree?

9. What is the height of a tree?

10. What is a parent? A child?

11. How does a parent differ from an ancestor?

12. What is an in-order traversal?

13. What is a pre-order traversal?

14. What is a level-order traversal?

15. What is the different between a depth first search and breadth first search?

Binary Search Trees:

1. What is a binary search tree?

2. How do you insert an item into a BST?

3. How do you remove an item from a BST?

4. What is treesort?

5. Briefly describe the treesort algorithm

6. What is the advantage of creating a balanced BST?

Hash Tables:

1. What is a hash table?

2. What is a hash function?

3. What properties are desirable for a hash function?

4. How is open addressing different from table restructuring?

5. What is a collision?

6. What is linear probing?

7. What is double hashing?

8. When might you want to use a hash table?