CS61B Spring 2015 Guerrilla Section 3 Worksheet

Leo Colobong, Nick Fong, Jasmine Giang, Laura Harker, Yujie Huang, Anusha Ramakuri, Nick Rose, Khalid Shakur, Jason Won, Fan Ye, Jason Zhang, Giulio Zhou

7 May 2015

Directions: In groups of 4-5, work on the following exercises. Do not proceed to the next exercise until everyone in your group has the answer and *understands why the answer is what it is*. Of course, a topic appearing on this worksheet does not imply that the topic will appear on the midterm, nor does a topic not appearing on this worksheet imply that the topic will not appear on the midterm.

1 Sorting with Heaps

Given an array of n elements, where each element is at most k away from the target position, sort it as efficiently as possible.

Hint: Consider using a priority queue and aim for a runtime of $O(n \log k)$.

```java
public void sortK(int arr[], int n, int k) {
}
```

STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section!

2 More Heaps

Design and code a data structure which supports adding **ints** in $O(\log n)$ time and finding the median in constant time, where n is the number of items in the data structure.

```java
import java.util.PriorityQueue;

public class MedianPQ {

    PriorityQueue<Integer> maxPQ; //Keeps track of the lower half
    PriorityQueue<Integer> minPQ; //Keeps track of the upper half

    public MedianPQ() {
        maxPQ = new PriorityQueue<Integer>();
        minPQ = new PriorityQueue<Integer>();
    }

    public void add(int x) {
















    }

    public Integer findMedian() {
















    }
}
```

STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section!

3 Even More Heaps

Consider a valid minheap h.

1. If we add a constant c to every value in h, is h still a valid minheap?

2. If we multiply every value in h with a constant c, is h still a valid minheap?

STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section!

4 Sorts

Execution times that result from applying sorting methods to sort sequences of 2000 values into ascending order appear in the table below. The algorithms possibly used are selection sort, insertion sort, and merge sort, and tree sort (repeated insertions into a binary search tree with no attempt to balance, followed by traversal of the tree). Specify which times go with which sorting method.

Time to sort 2000 random values	Time to sort 2000 values already in increasing order	Time to sort 2000 values already in decreasing order	Sorting Method (Selection, Insertion, Merge, or Tree)
326	2	597	
52	5596	5209	
422	423	419	

STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section!

5 Graphs

Youre given an undirected, weighted graph $G = (V, E)$, a list of start vertices, and a list of end vertices. Describe an efficient algorithm that returns the shortest path from some start vertex to some end vertex.

STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section!

6 More Graphs

Given an undirected graph $G = (V, E)$ and an edge $e = (u, v)$ in G, create an $O(V + E)$ time algorithm to determine whether G has a cycle containing e.

```java
public class CycleDetective {
















































}
```

STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section!

7 Even More Graphs

Given a connected, undirected, weighted, graph, give a strategy to construct a set with as few edges as possible such that if those edges were removed, there would be no cycles in the remaining graph and that the sum of the weights of the edges you remove is as big as possible. This strategy must be as quick as possible. List any data structures you might need and how they would be utilized.

STOP!

DON'T PROCEED UNTIL EVERYONE IN YOUR GROUP HAS FINISHED AND UNDERSTANDS ALL EXERCISES IN THIS SECTION!

8 Tree Traversals

1. Draw the binary tree that corresponds to the traversals:
   `in-order: 3 8 4 5 7 9 2 1`
   `post-order: 3 4 8 7 9 1 2 5`

2. Now suppose youre given the traversals for a full binary tree (i.e. each node has 0 or 2 children). Draw the corresponding tree:
   `pre-order: 8 6 7 5 3 9 2 1 4`
   `post-order: 5 3 7 9 6 1 4 2 8`

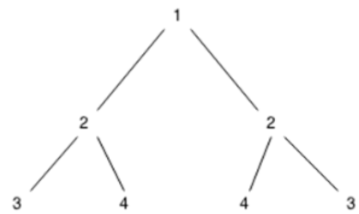
STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section!

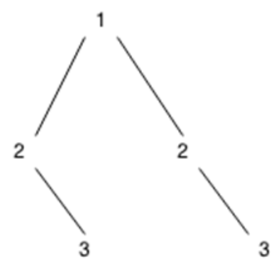
9 Even More Graphs

Given a binary tree, check whether it is a mirror of itself (i.e, symmetric around its center).

For example, this binary tree is symmetric:



But the following tree is not:



Given the following `TreeNode` implementation, fill in the `isSymmetric` method of the `Symmetric` class.

```java
public class TreeNode {
    int val;
    TreeNode left;
    TreeNode right;
    TreeNode(int x) {
            val = x;
    }
}

public class Symmetric {
    public boolean isSymmetric(TreeNode root) {

    }
}
```

STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section!

10 Hashing

1. What is the worst case runtime of a single call to insert on a hashtable with n items?
2. In what situations would we achieve the worst case runtime?