

CMSC132

Spring 2006

Midterm #2

First Name: _____

Last Name: _____

Student ID: _____

Section time _____ TA: _____

Grader Use Only:

#1 Hashing		(12)
#2 Networking		(12)
#3 Sorting		(18)
#4 Complexity		(12)
#5 Reg Exp		(10)
#6 Recursion		(12)
#7 Lists		(12)
#8 Maps/Sets		(12)
Total		(100)
Honors		(10)

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: _____

General Rules (Read):

- This exam is closed book and closed notes.
- If you have a question, please raise your hand.
- Total point value is 100 points.
- The short answer questions are not essay questions. Strive to answer them in 1 or 2 sentences. Longer answers are not necessary and are discouraged.
- **WRITE NEATLY.** If we cannot understand your answer, we will not grade it (i.e., 0 credit).
- **PUNT RULE:.** For any question, you may write PUNT, and you will get $\frac{1}{4}$ of the points for the question (rounded down). If you feel totally lost on a question, you are encouraged to punt rather than waste time writing down a bunch of vaguely related verbiage in hopes of getting some partial credit.
- **Honors section questions only count for credit for students in the honors section.**

Problem 1 Hashing/Other (12 pts)

1. (4 pts) In Java, what properties must a hashCode() function have? What can go wrong if it does not have this property?

2. (4 pts) What other properties are desirable for a hashCode() function? What can go wrong if it does not have this property?

3. (4 pts) What is the expected and worse case asymptotic running time of searching a hash table of n elements to see if it contains an element?

Problem 2 Networking (12 pts)

1. (3 pts) Does HTTP use UDP or TCP?
2. (3 pts) Say you have two computers on the network (machines A and B). Machine A needs to send a short, 4-byte message to machine B, and doesn't expect to see any message in response. Discuss the merits of using TCP vs. UDP for this communication.
3. (3 pts) Give the first line of a request that a web browser might send to the CNN web server (using the HTTP protocol) to display the page <http://www.cnn.com/index.html>
4. (3 pts) What is the task performed by the **accept** method of the Java ServerSocket class?

Problem 3 Sorting (18 pts)

1. (3 pts) What is the average asymptotic complexity of quicksort?
2. (3 pts) Name two quadratic sorting algorithms discussed in class.
3. (3 pts) Name a sorting algorithm has an average and worst case asymptotic complexity equals to $O(n \log(n))$?
4. (3 pts) Show a possible result of partitioning (as in quicksort) the array below, using the first element as the partition element:

12 4 30 10 8 20

5. (3 pts) What is a stable sorting algorithm? Briefly explain.
6. (3 pts) What is an in-place sorting algorithm? Briefly explain

Problem 4 Algorithmic Complexity (12 pts)

1. (3 pts) What is the best asymptotic worst case time complexity for searching for an element in an array of n elements? What if the algorithm only needs to handle sorted arrays?

2. (3 pts) Sort the following complexity categories from least to most complex.

$O(\log(n))$

$O(n^2)$

$O(n \log(n))$

3. (6 pts) Calculate the asymptotic complexity of the code snippets below (using big-O notation) with respect to the problem size n :

- a. $f(n) = O($)

```
int i=n-4;
while (i++ < n) {
    int k = 0;
    while (k++ < n)
        System.out.println("Bye");
}
```

- b. $f(n) = O($)

```
for (int k=0; k<n; k += n/2)
    for (int m=1; m<=n && m <= 30; m++)
        System.out.println("Hello");
```

- c. $f(n) = O($)

```
for (int k=0; k<n; k++)
    for (int m=1; m<=n; m++)
        System.out.println("Hello");
for(int k = 0; k < 10*n; k++)
    System.out.println("Bye");
```

Problem 5 Regular Expressions (10 pts)

1. (3 pts) Write a Java regular expression for the language that includes strings that result from concatenating a single lowercase letter, a single digit and letter A. Examples of strings in the language are:

b5A

m6A

e2A

2. (7 pts) What is the language defined by the following Java regular expression? Describe it in plain English, and then provide 4 test cases of strings in the language, and 4 test cases of strings of a's, b's, and c's that are not in the language.

(alb)+c+

Problem 6 Recursion (10 pts)

1. (10 pts) For this problem you will implement a recursive method named **onlyDigits** that returns a string with only the digits (if any) present in the parameter. The method has the following prototype:

```
public static String onlyDigits(String str);
```

Non-recursive implementation will receive no credit. For example, you may not use any loops or regular expression methods in your implementation of **onlyDigits**. You **may also not** define an auxiliary method to implement this problem. The following methods could prove helpful:

boolean Character.isDigit(char c) – Static method that returns true if the argument is a digit.

String String.charAt(int pos) – Non-static String method that returns the character at the specified index value.

String String.substring(int pos) – Non-static String method that returns the substring starting at the specified index value.

To clarify any confusion about what **onlyDigits** should do, here are some test cases:

```
public void test() {  
    assertEquals("123", OnlyDigits.onlyDigits("123"));  
    assertEquals("123", OnlyDigits.onlyDigits("x1x2x3x"));  
    assertEquals("", OnlyDigits.onlyDigits("xyxya xks ska; isjksd lk"));  
}
```

Problem 7 Linear Data Structures (12 pts)

1. The following Java class definition is for a singly linked list:

```
public class LinkedList<E> {  
    private Node<E> head;      // Represents the first node of the list  
    // ADDITIONAL INSTANCE VARIABLES YOU MAY NEED GO HERE  
  
    class Node<E> {  
        E data;  
        Node<E> next;  
    }  
}
```

Define a method named **add** which appends an element at the *end* of the list. Feel free to add any instance variables you may need. If the constructor needs to initialize any information, provide a constructor. Give the asymptotic efficiency of your add operation as a function of the length of the list (your add function doesn't have to be as efficient as possible).

Problem 8 Maps/Sets (12 pts)

Consider the following class. The main method is primarily provided to ensure that you understand how mapWords will be called; you can probably understand and answer the question without looking at the main method, but it is provided as a backup/auxiliary explanation.

```
public class MapWords {  
  
    public static void main(String[] args) throws IOException {  
        Map<String, Set<Integer>> map = new HashMap<String, Set<Integer>> ();  
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));  
        int line = 1;  
        String s;  
        while((s = in.readLine()) != null)  
            mapWords(map, s.split(" "), line++);  
        System.out.println(map);  
    }  
  
    /** update map by adding lineNumber to Set corresponding to each  
        String in the array strings; note that if a string has never been seen before, it  
        will be mapped to null rather than an empty set */  
    static void mapWords(Map<String, Set<Integer>> map,  
                        String[] strings, int lineNumber) {  
        ... }  
    }
```

For example, when run on the input:

```
a b c  
d e f  
a d f
```

The class might generate the output

```
{d=[2, 3], a=[1, 3], c=[1], f=[2, 3], b=[1], e=[2]}
```

1. (3 pts) Suppose we use a TreeMap rather than a HashMap above. What is the difference in output we will see when we try to print the map?

2. (3 pts) Which of the following map methods will allow you to get a collection of all the words in the map? Circle your choice.
 - a. `keySet`
 - b. `values`
 - c. `hashCode`
3. (8 pts) Provide an implementation of the method `mapWords`. Remember that the **get** method allows us to retrieve an element from the map and the **put** method allow us to insert/update an element in the map.

Honor Section (10 pts)

1. (5 points) You are studying a problem X, and I don't think you can develop an efficient algorithm for it. You think problem X is related to graph coloring, which is an NP-complete problem. Sketch, very briefly, how you might be able to find a way to use the relationship between X and graph coloring to show that it is unlikely that you can develop an efficient algorithm for X.

2. (5 points) You should be familiar with the Fibonacci numbers 1, 1, 2, 3, 5, 8, 13,, where $\text{fib}(1) = \text{fib}(2) = 1$ and $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$.

Define a class `FibonacciIterator` that implements `Iterator<Integer>`. Using this iterator will iterate over the Fibonacci numbers. The Iterator should not stop at any point, but just continue generating Fibonacci numbers (you can ignore the fact that eventually you will get overflow with 32 bit integers).