

COP 3530
Data Structures

Midsemester Exam

Name: _____

February 23, 2012

This exam has 4 questions. Each question starts on a new page. Please answer each question on its page. You may assume `java.util` has been imported. There will be no deductions for lack of commenting. There will be no deductions for lack of import directives. There will be no deductions for minor syntax errors.

1. [50 points]

Big-Oh multiple choice questions not shown.

2. [50 points] This question requires that you implement some methods for a class that represents a doubly-linked list. In this question, **both a header and a tail are used**. You may assume an appropriate declared nested class `Node`. You may assume that the list does not store `null` values. You should only be following links; your solutions should not create or use any iterator classes.

(a) Below you will implement `toString`, `removeFirst`, and `addLast`. Before writing the code, give the Big-Oh running time for each routine.

(b) Implement `toString`. You may not invoke other methods of this class.

```
public String toString( )  
{
```

```
}
```

(c) Implement `remove` below. You may not invoke any other methods of the class. Be careful to correctly handle empty lists.

```
public void removeFirst( )  
{
```

```
}
```

(d) Implement `addLast` below. You may not invoke any other methods of the class.

```
public void addLast( )  
{
```

```
}
```

DID YOU REMEMBER TO GIVE THE BIG-OH?

3. [50 points] A `StringSet` stores strings and supports the following three operations: `add`, `contains`, and `containsCaseInsensitive`. Duplicates are not allowed. If "HELLO" and "Hello" are both added to a `StringSet s`, then `s.contains("hello")` will be false, but `s.containsCaseInsensitive("hello")` will be true, as it matches two strings in the `StringSet s`.

A `StringSet` can be implemented by using a map, in which the key is the lower case representation of a string, and the value is a set of strings that have been added into the map that have the corresponding representation. In the above example, the key would be "hello" and the corresponding value would be ["HELLO", "Hello"].

Implement the missing components of `StringSet` below. You can use `str.toLowerCase()` to compute `str`'s lower case equivalent.

```
class StringSet
{
    public void add( String str )
    {

    }

    public boolean contains( String str )
    {

    }

    public boolean containsCaseInsensitive( String str )
    {

    }

    private Map<String,Set<String>> theMap = new TreeMap<>( );
}
```

4. [50 points] Implement a method, `findMax`, that returns the largest file in a directory. You must include all subdirectories in your search.

The `File` class has the following useful methods:

```
boolean isDirectory( );
long length( );
File [ ] listFiles( );
```

Implement `findMax` below.

```
// If d is a regular file, return it.
// If d is a directory, search the directory and all subdirectories
// and return the File with the largest size.
private static File findMax( File d )
{
```