SER 222 **Practice** Exam 2

Updated 2/3/2019

Last Name: ____________________________
First Name: ____________________________
Last 4 digits of ASU ID: ____________________

Exam Instructions

The exam is open textbook (Algorithms 4e by Sedgewick and Wayne), as well as open note. No electronic items are allowed. Write legibly. Please use a pen (instead of a pencil) if you have one. There are 110 points available and the exam must be completed in 37.5 minutes. This exam has two types of questions:

Short answer questions: There are 80 points of short answer questions. A typical answer is one or two sentences. Each short answer question is worth 10 or 5 points.

Write-in questions: The programming questions are given near the end of the paper. They must be answered on the question paper. There are 0 points of write-in programming questions.

Topic	**Earned**	**Possible**
SA: Elementary Sorts		20
SA: Mergesort		20
SA: Priority Queues		20
SA: Symbol Tables		20
Total:		80

Short Answer: Elementary Sorts

1. What are the Big-Oh and Omega orders of the following code fragment? The fragment is parametrized on the variable n. Assume that you are measuring the number of swap calls. [10 points]

```
public static void sort(Comparable[] a) {
     int n = a.length;
     for (int j = 0; j < n-1; j++) {
          int z = j;
          for (int i = j+1; i < n; i++) {
               if (a[i] < a[z]) {
                    z = i;
               }
          }

          if(z != j) {
            swap(a[j], a[z]); //count these
          }
}
```

   (a) What is the Big-Oh order of the above code fragment? If it does not exist, then explain.

   (b) What is the Tilde order of the above code fragment? If it does not exist, then explain.

2. When dealing with systems having low RAM capacity, or analyzing large datasets, space is at a premium. In these cases, algorithms must be designed to reduce their memory foot print. From the sorting algorithm implementations seen in class, which would be the best choice? [10 points]

Short Answer: Mergesort

3. [Acuña] Consider the following array: 23, 7, 35, 3, 4, 2, 13. Show a trace of execution for top-down mergesort. Illustrate how the array is broken down, and then merged into an ordered state. [10 points]

4. In the lower bound proof for sorting, why must there be at least N! leaves on the decision tree? [10 points]

Short Answer: Priority Queues

5. What is the difference between a (max) heap and a priority queue? [10 points]

6. One of the main operations for a PQ is to move an element lower in a heap, so that it will be in the proper order. It is implemented by the sink() method. According to lecture, this particular implementation takes $O(log(n))$ number of exchanges to put the element in its proper place.

```
private void sink(int k) {
    while (2*k <= N) {
        int j = 2*k;
        if (j < N && less(j, j+1))
          j++;
        if (!less(k, j))
          break;
        exch(k, j);
        k = j;
    }
}
```

Support this claim by explaining why this method is $O(log(n))$: [10 points]

Short Answer: Symbol Tables

7. Trace an initially empty symbol table (called ST) through the following operations:

```
SymbolTable<> ST = new BinarySearchTree<String, Integer>();
ST.put("SER100", 150);
ST.put("SER200", 150);
ST.put("SER250", 100);
ST.put("SER316", 100);
System.out.println(ST.get("SER200"));
ST.put("SER200", 250);
ST.put("SER316", 75);
```

Give the contents of the hashtable after the code has been executed. Use the format ABC### : ### (e.g., "SER100 : 100") to give your answer. Use separate lines for each key/value pair. [10 points]

8. Consider the task of implementing a SymbolTable vs an OrderedSymbolTable. For the purposes of get and put, which you be more likely to create an efficient implementation for? Explain. (Hint: no knowledge of BSTs or hashtables is needed.) [10 points]

Programming: ???

There will be at least one programming question on the actual exam.