COS 226	Algorithms and Data Structures	Fall 2005

Midterm

This test has 6 questions worth a total of 50 points. You have 80 minutes. The exam is closed book, except that you are allowed to use a one page cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. **Write out and sign the Honor Code pledge before turning in the test.**

"I pledge my honor that I have not violated the Honor Code during this examination."

Problem	Score
1	
2	
3	
Sub 1	

Problem	Score
4	
5	
6	
Sub 2	

Total	

Name:

Login ID:

Precept: 1 12:30 Keith
3 3:30 Harlan

1. **8 sorting algorithms. (8 points)**

   The column on the left is the original input of strings to be sorted. The columns to the right are the contents at some intermediate step during one of the 8 sorting algorithms listed below. Match up each algorithm by writing its number under the corresponding column. Use each number exactly once.

Jane	Adam	Anna	Abby	Will	Adam	Jada	Abby	Adam	Abby
Adam	Alex	Adam	Cole	Seth	Alex	Emma	Adam	Dave	Adam
Mary	Cole	Abby	Alex	Ryan	Abby	Ella	Alex	Erik	Alex
Jeff	Dave	Ella	Anna	Sean	Anna	Maya	Anna	Erin	Anna
Erik	Erik	Emma	Adam	Mark	Cole	Anna	Cole	Evan	Cole
Dave	Erin	Dave	Dave	Noah	Dave	Sara	Dave	Jack	Dave
Evan	Evan	Alex	Erin	Owen	Erik	Eric	Ella	Jada	Ella
Sean	Jack	Cole	Emma	Sara	Evan	Jane	Emma	Jane	Emma
Erin	Jada	Eric	Ella	Hart	Erin	Dave	Eric	Jeff	Eric
Jada	Jane	Jada	Eric	Joey	Emma	Luke	Erik	Mary	Erik
Jack	Jeff	Jack	Erik	Jack	Ella	Kyle	Erin	Noah	Erin
Noah	Kyle	Noah	Evan	Maya	Eric	Cole	Evan	Sean	Evan
Luke	Luke	Luke	Luke	Luke	Hart	Jake	Luke	Luke	Hart
Kyle	Mary	Kyle	Kyle	Kyle	Jane	Jeff	Kyle	Kyle	Jack
Owen	Noah	Owen	Owen	Mary	Jeff	Noah	Owen	Owen	Jada
Seth	Owen	Seth	Seth	Jeff	Jada	Seth	Seth	Seth	Jake
Cole	Sean	Sean	Sean	Eric	Jack	Leah	Jada	Cole	Jane
Alex	Seth	Evan	Sara	Alex	Joey	Josh	Mary	Alex	Jeff
Hart	Abby	Hart	Hart	Erin	John	Erik	Hart	Hart	Joey
Mark	Anna	Mark	Mark	Jada	Josh	Jack	Mark	Mark	John
Joey	Ella	Joey	Joey	Erik	Jake	Mark	Joey	Joey	Josh
Emma	Emma	Erik	Will	Emma	Kyle	Will	Sean	Emma	Kyle
Ella	Eric	Jeff	Jeff	Ella	Luke	Adam	Noah	Ella	Leah
Lily	Hart	Lily	Lily	Lily	Lily	Evan	Lily	Lily	Lily
Maya	Jake	Maya	Maya	Dave	Leah	Sean	Maya	Maya	Luke
Leah	Joey	Leah	Leah	Leah	Mary	Erin	Leah	Leah	Mark
Abby	John	Mary	Mary	Abby	Mark	Owen	Jane	Abby	Mary
Anna	Josh	Jane	Jane	Anna	Maya	John	Jeff	Anna	Maya
John	Leah	John	John	John	Noah	Ryan	John	John	Noah
Ryan	Lily	Ryan	Ryan	Evan	Owen	Hart	Ryan	Ryan	Owen
Josh	Mark	Josh	Josh	Josh	Ryan	Alex	Josh	Josh	Ryan
Jake	Maya	Jake	Jake	Jake	Sean	Mary	Jake	Jake	Sara
Sara	Ryan	Sara	Noah	Jane	Seth	Joey	Sara	Sara	Sean
Will	Sara	Will	Jack	Cole	Sara	Lily	Will	Will	Seth
Eric	Will	Erin	Jada	Adam	Will	Abby	Jack	Eric	Will
----	----	----	----	----	----	----	----	----	----
0									

(0) Original input

(1) 3-way radix quicksort

(2) Heap sort

(3) Insertion sort

(4) LSD radix sort

(5) Mergesort

(6) MSD radix sort

(7) Quicksort

(8) Selection sort

(9) All of them

2. **Algorithm Properties. (10 points)**

Match up each *worst-case* quantity on the left with the best matching asymptotic value on the right. You may use a letter more than once.

___ Max height of a binary heap with N items.	A. 1
___ Max height of red black tree with N items.	B. $\log^* N$
___ Max function call stack depth to mergesort N items.	C. $\log N$
___ Max number of probes to search for a key in a double hashing table with N key-value pairs.	D. N
___ Max height of a WQUPC (weighted quick union with path compression) tree with N items.	E. N^2

3. **Analysis of algorithms. (6 points)**

Each of the Java functions on the left take a string s as input, and returns its reverse. Choose the best matching asymptotic complexity (as a function of the string length N) bound on the right. Recall that concatenating two strings in Java takes time proportional to the sum of their lengths, and extracting a substring takes constant time.

___
```
public static String reverse1(String s) {
   int N = s.length();
   String reverse = "";
   for (int i = 0; i < N; i++)
      reverse = s.charAt(i) + reverse;
   return reverse;
}
```

___
```
public static String reverse2(String s) {
    int N = s.length();
    if (N <= 1) return s;
    String left = s.substring(0, N/2);
    String right = s.substring(N/2, N);
    return reverse2(right) + reverse2(left);
}
```

___
```
public static String reverse3(String s) {
    int N = s.length();
    char[] a = new char[N];
    for (int i = 0; i < N; i++)
       a[i] = s.charAt(N-i-1);
    return new String(a);
}
```

A. $\log N$

B. N

C. $N \log N$

D. N^2

E. 2^N

4. **Priority queues. (8 points)**

   Insert the following keys into an initially empty *minimum-based* binary heap.

   ```
   A L G O R I T H M
   ```

   Show the heap representation (the array) and the binary tree representation (the picture).

   Now perform a *delete-the-minimum* operation and show the resulting binary tree representation (the picture).

5. **Red-black trees. (8 points)**

   Draw the top-down 2-3-4 tree that results when the keys

   ```
   A L G O R I T H M
   ```

   are inserted in that order into an initially empty tree, using the standard top-down algorithm. Then *draw* the corresponding red-black tree that is constructed by the standard top-down red-black tree algorithm.

6. **Longest common substring. (10 points)**

   Your must find the longest (contiguous) substring that appears in both Tolstoy's *War and Peace* and Sedgewick's *Algorithms in Java*. How would you write a program to compute the answer as quickly as possible? Describe and justify your approach. Your solution will be graded on correctness, efficiency, and clarity.