Homework 2: Stacks and Sorting

Handed out: **September 25, 2012**
Due: **October 2, 2012**

Submission

Submit your answers to the questions below **as a PDF document**. Failure to do so will result in points loss. If you do not know how to generate a PDF document, do ask! You will use turnin for this submission and name your PDF file <your_first_name>_<your_last_name>.pdf, as you did for the first project. The turnin command will therefore be:

```
% turnin -c cs251 -p homework2 <your_first_name>_<your_last_name>.pdf
```

Grading

There are a total of 100 points for this homework, distributed as follows.

Question 1. (15 points)

Match each of the sorting algorithms below with its primary distinguishing characteristics (aspresented in lecture and in the book) by writing the combinations of letter and number corresponding to each algorithm and its characteristic. You should use each letter once and only once.

A. Mergesort	*1. Works well with duplicates*
B. Quicksort	*2. Works well with order*
C. Shellsort	*3. Not analyzed*
D. Insertion sort	*4. Stable and fast*
E. Selection sort	*5. Optimal data movement*
F. 3-way quicksort	*6. Fast general-purpose sort*

Question 2. (20 points)

```
int count = 0;
int N = a.length; Arrays.sort(a);
for (int i = 0; i < N; i++) {
   for (int j = i+1; j < N; j++){
      if (Arrays.binarySearch(a, a[i] + a[j])) count++;
   }
}
```

Consider the code fragment above. Suppose it takes 1 second to run the code when N = 3500. Approximately how long will it take when N = 35000? Circle the best answer.

10 seconds *20 seconds* *1 minute* *2 minutes* *1 hour* *2 hours*

Question 3. (15 points)

Prove that the number of compares used by mergesort is monotonically increasing (C(N+1 > C(N) for all N>0).

Question 4. (15 points)

Given 3 lists of N names each, propose an algorithm to determine if there is any name common to all three lists, and if so, return the first such name. Your algorithm must have O(N log N) complexity.

Question 5. (15 points)

Suppose that an intermixed sequence of (stack) push and pop operations are performed. The pushes push the integers 0 through 9 in order; the pops print out the return value. Which of the following printed sequence(s) could not occur? Indicate any impossible sequence(s).

(a) 4 3 2 1 0 9 8 7 6 5
(b) 4 6 8 7 5 3 2 9 0 1
(c) 2 5 6 7 4 8 9 3 1 0
(d) 4 3 2 1 0 5 6 7 8 9

Question 6. (20 points)

An array is *bitonic* if it is comprised of an increasing sequence of integers followed immediately by a decreasing sequence of integers. Describe the steps of an algorithm that, given a bitonic array of N distinct int values, can determine whether a given integer is in the array. Your algorithm should use ~ 3 log N compares in the worst case.