

Name: \_\_\_\_\_ Andrew Id: \_\_\_\_\_

**15-121 Fall 2019 Quiz 8**

Up to 25 minutes. No calculators, no notes, no books, no computers. Show your work!

**1. Short Answer**

(a) (1 point) Name an  $O(N^2)$  (worst case) sort.

(b) (1 point) Name an  $O(N \log N)$  (worst case) sort.

(c) (3 points) What does it mean for a sort to be stable?

(d) (3 points) Consider the partition operation from Quick Sort. Assuming that the first element in the following array is your pivot, give a valid order for the array after the partition operation is performed on it. Note: Sorting the array is not an acceptable solution. You should simply partition it with the appropriate pivot.)

[68, 52, 85, 22, 94, 50, 12, 32, 73]

(e) (4 points) Assume I am building a simple calculator that has the following behavior. When I see an integer, I push its value onto a stack. When I see an operator (+, -, \*, /), I pop the stack once, and store the result in a variable called `op2`. I then pop the stack a second time, and store the result in a variable called `op1`; I then perform the arithmetic operation specified by that operator (`op1 operator op2`) and push the result of that operation back onto the stack.

For example, if I entered `7 2 -`, `peek()` would return 5 since 7 would be pushed onto the stack, then 2, and then the `-` would cause 2 to be popped and stored in `op2`, 7 to be popped and stored in `op1`, and the result of `7-2`, which is 5, would be pushed onto the stack.

Tell me what value is returned by `peek()` after the following sequence of values is processed. Write your final answer in the box, but show you work in the empty space next to the box.

`5 2 * 6 2 / -`

## 2. (8 points) Code Tracing

Indicate what the following program prints. Place your answer (and nothing else) in the box under the code. Note that the first two lines of output have been provided for you so that you can see the proper format and order when printing a `MyStack` or `MyQueue`.

```
public class CT {
    public static void main(String[] args) {
        MyStack<String> s = new MyStack<String>();
        MyQueue<String> q = new MyQueue<String>();

        s.push("A");
        s.push("B");
        s.push("C");
        System.out.println("1: " + s);

        q.enqueue("D");
        q.enqueue("E");
        q.enqueue("F");
        System.out.println("2: " + q);

        while (!s.isEmpty()) {
            String t = s.pop();
            q.enqueue(t);
        }
        System.out.println("3: " + s);
        System.out.println("4: " + q);

        q.enqueue("G");
        s.push("H");
        System.out.println("5: " + s);
        System.out.println("6: " + q);

        while (!q.isEmpty()) {
            String t = q.dequeue();
            s.push(t);
        }
        System.out.println("7: " + s);
        System.out.println("8: " + q);

        while (!s.isEmpty()) {
            String t = s.pop();
            q.enqueue(t);
        }
        System.out.println("9: " + s);
        System.out.println("10: " + q);
    }
}
```

```
1: [A, B, C]
2: [D, E, F]
```