CS 61B Discussion 1 Fall 2017

1 Javaian Rhapsody

Next to each line, write out what you think the code will do when run. (Assume the `Singer` class exists and that the code below compiles.)

```
1   String disagree = "no";
2   int x = 7;
3   Singer queen = new Singer("Queen");
4
5   while (x > 0) {
6     x -= 1;
7     queen.sing(disagree);
8   }
9
10  String[] phrases = {"Oh", "mamma mia", "let me go"};
11  System.out.print(phrases[0]);
12  for (int i = 0; i < 3; i += 1) {
13    System.out.print(" " + phrases[1]);
14  }
15  System.out.print(" " + phrases[2]);
```

2 Fibonacci

Implement this function recursively.

```
/** fib(N) returns the N^th Fibonacci number, for N >= 0.
  * The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13, 21, ... */
public static int fib(int N) {
```

Now implement a function that provides the same results but avoids redundant computations:

```
public static int fib2(int N) {
```

Now use this modified function header to write a recursive solution that avoids redundant computations and that has a body that is 5 lines or fewer. The inputs are defined as follows: your goal is to compute the `N`th fib number, you are currently computing the `k`th fib number (which is `f0`), and the `(k+1)`th fib number is `f1`.

```
public static int fib3(int N, int k, int f0, int f1) {
```

3 Mystery

```
1   public static int mystery(int[] inputArray, int k) {
2       int x = inputArray[k];
3       int answer = k;
4       int index = k + 1;
5       while (index < inputArray.length) {
6           if (inputArray[index] < x) {
7               x = inputArray[index];
8               answer = index;
9           }
10          index = index + 1;
11      }
12      return answer;
13  }
14
15  public static void mystery2(int[] inputArray) {
16      int index = 0;
17      while (index < inputArray.length) {
18          int targetIndex = mystery(inputArray, index);
19          int temp = inputArray[targetIndex];
20          inputArray[targetIndex] = inputArray[index];
21          inputArray[index] = temp;
22          index = index + 1;
23      }
24  }
```

- What does `mystery` return if `inputArray` is the array {3, 0, 4, 6, 3}, and `k` is 2?

- Describe, in English, what `mystery` returns.

- Extra for experts: What does `mystery2` return if `inputArray` is the array {3, 0, 4, 6, 3}? Then, describe, in plain English, what `mystery2` does to the array.