CS61B Week 6: Bit Business & Algorithmic Analysis

Bit Business

1. Write 47 in binary.
2. Write 463 in hexadecimal.

For the next 4 problems assume we are working with byte-length numbers.

3. Write `0b11001001` in decimal. What is `0b11001001 >> 2`? What is it in decimal?
4. Write `0xC9` in decimal. What is `0xC9 >>> 2`? What is it in decimal?
5. What is `23 << 2`? What about `23 << 3`? `23 << 4`?
6. What is `35 >> 2 << 2`?
7. Assume a, b, and c are integers. What is `(((a | b) & a) ^ c) ^ c`?

Algorithmic Analysis

1. What is the big-Theta running time of the following code in terms of n?

```
public static void printStuff(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++) {
            System.out.println(i + j);
        }
    }
}
```

2. Assume sortedList is a sorted list of length n with no duplicates. What is the running time of the function useless? What does it print?

```
static void useless(int[] sortedList) {
    for (int i = 0; i < sortedList.length; i++) {
        System.out.println(foo(sortedList, sortedList[i]));
    }
}

static int foo(int[] lst, int toFind) {
    return bar(lst, toFind, 0, lst.length);
}

static int bar(int[] lst, int toFind, int lower, int upper) {
    if (lower == upper) {
        return -1;
    }
    int mid = (lower + upper) / 2;
    if (lst[mid] > toFind) {
        return bar(lst, toFind, lower, mid);
    } else if (lst[mid] < toFind) {
        return bar(lst, toFind, mid + 1, upper);
    }
    return mid;
}
```

3. The following function returns the nth fibonacci number. What is its running time?

   ```
   public static void fib(int n) {
       if (n == 1 || n == 0)
           return 1;
       return fib(n - 1) + fib(n - 2);
   }
   ```

4. Rewrite the fibonacci function to use iteration instead of recursion. What is the new running time?

   ```
   public static void iterativeFib(int n) {
   ```

Sample Midterm Question of the Week:
What does the following code print?

```
public class What {
    public long n;
    public void increment() {
        n++;
    }
    public static void reset(What w) {
        w.increment();
        w = new What();
        w.n = 0;
    }
    public static void main(String[] args) {
        What w = new What();
        w.n = 7;
        reset(w);
        System.out.println("The number is " + w.n);
    }
}
```

Sample Interview Questions of the Week:

1.
   ```
   int temp = b;
   b = a;
   a = temp;
   ```

   The above code snippet swaps the value of two integers, a and b. Write code to swap them without using a temporary variable.

2. You're given an array of integers of length n such that only one of the elements in the array appears once, while every other element appears exactly twice. How can you figure out the value of this one element that appears only once in O(n) time and O(1) space? (O(1) space means you use a constant number of variables no matter how long the given array is)