COS126/128: General Computer Science	Spring 1996

Exercise # 3

Handed out:March 5, 1996

These exercises cover the material on recursion and quicksort. You *will not* be held responsible for this material on the first exam. You *will* be held responsible for the material on a later exam. These exercises are meant to guide class discussion and to help you prepare for exams. *Do not* turn in solutions.

1. Give a reasonable recursive program for computing 2^N.
2. (Optional material.) Prove by induction that $1 + 2 + 4 + 8 + \ldots 2^N = 2^{N+1} - 1$.
3. Rewrite the binary search given in lecture

   ```
   int find (int A[], int lb, int ub, int v);
   ```

   so that if `v` *is not* in A, `find` returns -1, and if `v` *is* in A, `find` returns an index `i` such that `A[i] = v`.
4. What is the value of f(5), when f is defined as follows?

   ```
   int f(int N)
     {
       if (N == 0) return 1;
       return 2*f(N-1) - 1;
     }
   ```
5. What are the recursive calls made by the Quicksort implementation given in class when it sorts the file 4 7 3 5 2 1 6 ?
6. Given an example of a file with 7 element for which Quicksort sorts in the slowest manner possible. How many recursive calls are made in this worst case? How many times is each element examined in this worst case?