

## CSC282 Fall 2005 Homework #1

1. CLRS 2-2 (correctness of bubblesort).
2. Show (hint: there are three cases, and you need to remember  $0! = 1$ ).

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

3. Sequential Search uses the following algorithm:

```
int seqSearch(int[] E, int n, int K)
1   int ans, index;
2   ans = -1; // assume failure
3   for (index = 0, index <n, index ++)
4       if (K == E[index])
5           ans = index; // Success!
6           break; // go home.
// Continue loop
7   return ans;
```

Give a formula for the total number of operations done by the Sequential Search algorithm in the worst case for an array with  $n$  entries. Count comparisons of  $K$  with array indices, comparisons with the variable `index`, additions, and assignments to `index`.

4. List these functions from lowest to highest asymptotic order, if any two or more are of the same asymptotic order, indicate which. Be careful with the second set of functions. Assume  $0 < \epsilon < 1$ .

$$n, 2^n, n \lg n, n^3, n^2, \lg n, n - n^3 + 7n^5, n^2 + \lg n.$$

$$e^n, \sqrt{n}, 2^{n-1}, \lg \lg n, \ln n, (\lg n)^2, n!, n^{1+\epsilon}.$$

5. Prove or give counterexample: For every positive constant  $c$  and every function  $f$  from nonnegative integers into nonnegative reals,  $f(cn) \in \Theta(f(n))$ . Hint: consider some fast-growing functions from previous problem.
6. Prove or give counterexample: For every function  $f$  from nonnegative integers into nonnegative reals,  $o(f) = O(f) - \Theta(f)$ , where of course “ $-$ ” is “set difference”.
7. Prove or give counterexample: For every function  $f$  from nonnegative integers into nonnegative reals, no function  $g$  is in both  $\Theta(f)$  and  $o(f)$ . That is  $\Theta(f) \cap o(f) = \emptyset$ .

8. The first  $n$  cells of the array  $E$  contain integers sorted in increasing order. The remaining cells all contain some very large integer we may think of as infinity, called `maxint`. The array may be arbitrarily large (think of it as infinite), and *you don't know  $n$* . Give an algorithm to find the position of a given integer  $x$  ( $x < \text{maxint}$ ) in the array in  $O(\log n)$  time. (Turns out this is a useful technique in certain arguments about NP-complete problems).

9. The expression  $x \ln x$  often needs to be evaluated at  $x = 0$ , but  $\ln(0) = -\infty$ , so it isn't obvious what that value might be. Note that  $-x \ln(x)$  is positive for  $0 < x < 1$ . Show that  $-x \ln x$  approaches 0 as  $x$  approaches 0 through positive values. (Hint: apply easy variable rewrite aiming for our favorite French ratio-limit technique.)

10. CLRS 3-4, parts d and e.