CS 561, HW2

Prof. Jared Saia, University of New Mexico

Due: Sept. 26th

Please use no outside references in solving these problems.

1. Consider the recurrence $T(n) = 2T(n/2) + \log^2 n$

   (a) Use the Master method to get a general solution to this recurrence.

   (b) Now use annihilators (and a transformation) to get a tight upper bound on the solution to this recurrence. Show your work. (Note that your two bounds should match)

2. Consider the following function:

```
int f (int n){
  if (n==0) return 3;
  else if (n==1) return 5;
  else{
    int val = 2*f (n-1);
    val = val - f (n-2);
    return val;
  }
}
```

   (a) Write a recurrence relation for the *value* returned by f. Solve the recurrence exactly. (Don't forget to check it)

   (b) Write a recurrence relation for the *running time* of f. Get a tight upperbound (i.e. big-O) on the solution to this recurrence.

3. Exercise 6.1-4

4. Exercise 6.1-5

5. Exercise 6.4-2
6. Exercise 6.4-3
7. Exercise 6.5-5
8. Problem 6-3, parts (a) through (e)
9. Exercise 7.1-2
10. Problem 7-1
11. Problem 7-2
12. Problem 7-3