CS 361, HW3

Prof. Jared Saia, University of New Mexico

Due: March 13, 2003

You are free to work with your group, or use any book or the web as a resource in doing this homework assignment. However, you must write up the work yourself. *In the first three questions, assume $T(n)$ is a constant for $n \leq 2$.*

1. Consider the recurrence $T(n) = 2T(n/2) + n^3$
   (a) Use the recurrence tree method to get a tight upper bound (i.e. big-O) on the solution to this recurrence
   (b) Now use annihilators (and change of variables) to get a tight upper on the solution to this recurrence. (your two bounds should match)
2. Consider the recurrence $T(n) = 16T(n/4) + n^2$
   (a) Use the recurrence tree method to get a tight upper bound (i.e. big-O) on the solution to this recurrence
   (b) Now use annihilators (and change of variables) to get a tight upper on the solution to this recurrence. (your two bounds should match)
3. Consider the recurrence $T(n) = 2T(n/4) + 1$
   (a) Use the recurrence tree method to get a tight upper bound (i.e. big-O) on the solution to this recurrence
   (b) Now use annihilators (and change of variables) to get a tight upper on the solution to this recurrence. (your two bounds should match)
4. Consider the following function:

```
int f (int n){
  if (n==0) return 0;
  else if (n==1) return 1;
  else{
    int val = 4*f (n-1);
    val = val - 4*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.

5. CLRS Exercise 6.4-2 (prove the correctness of HeapSort)