

CS251 Data Structures – Fall 2011

Homework 5 – 55 points

Due: Nov. 7

1. (10 points) Consider the following recurrence relation

$$\begin{aligned} T(0) &= 8 \\ T(1) &= 17 \\ T(n) &= 5T(n-1) - 6T(n-2), \quad n > 1 \end{aligned}$$

- Determine $T(2), \dots, T(5)$.
- Using techniques discussed in class, solve the recurrence relation exactly. Show that your formula gives the correct value for $T(5)$.
- What Big-O class does this solution belong in?

2. (10 points) Consider the following recurrence relation

$$\begin{aligned} T(0) &= 8 \\ T(1) &= 17 \\ T(n) &= 5T(n-1) - 6T(n-2) + 4, \quad n > 1 \end{aligned}$$

- Determine $T(2), \dots, T(5)$.
- Using techniques discussed in class, solve the recurrence relation exactly. Show that your formula gives the correct value for $T(5)$. NOTE: Feel free to use the results of the previous problem where appropriate.

3. (10 points) Consider the following recurrence relation

$$\begin{aligned} T(0) &= 8 \\ T(1) &= 17 \\ T(n) &= 5T(n-1) - 6T(n-2) + 2n + 1 \end{aligned}$$

- Determine $T(2), \dots, T(5)$.
- Solve the recurrence relation exactly. Show that your formula gives the correct value for $T(5)$. HINT: Let $T(n) = S(n) + an + b$, and select a and b such that the n term and the constant term drop out of the recurrence relation. Then solve for $S(n)$, and in turn, $T(n)$.

4. (25 points) In class we studied the (n, m) string problem which counted the number of strings of length n which did not have more than m 1's in a row anywhere in the string. Let's extend this problem so that there are no more than m 1's or m 0's in the string. Let $T_m(n)$ = the number of strings which satisfy this new (n, m) problem.

- Determine $T_2(5)$ and $T_3(5)$.
- What is $T_1(n)$ for all n ? What is $T_{n-1}(n)$ for all n ?
- Write a branch-and-bound algorithm which computes $T_m(n)$.
- Develop a recurrence relation (with initial conditions) for $T_2(n)$. HINT: This one is a bit tricky. Use a tree diagram showing how strings can be built. Also, note that exactly half of the strings in $T_2(n)$ start with a 0, and half start with a 1.
- Solve your recurrence relation to get a closed form for $T_2(n)$. Use this to determine $T_2(20)$.