5DV022 Problem Exercise 1 Fall 2008

Solutions to this assignment are due on September 11, 2008 at 1700 (5pm). The signed cover sheet must be turned in with your solutions.

Turn your solutions in to the course instructor. Do not put them in the mailboxes for laboratory reports.

1. Suppose that you are working with an algorithm whose time complexity $T(n)$ on an instance of size n is known to be governed by the following homogeneous recurrence equality.

$$T(n+2) = 15 \cdot T(n+1) - 56 \cdot T(n)$$

The initial conditions are $T(0) = 5$ and $T(1) = 38$.

(a) Give the characteristic polynomial of the associated recurrence relation.

(b) Solve for the roots of this polynomial.

(c) Give the solution to the recurrence equation.

(d) Give the best upper bound on the complexity of the algorithm based upon this solution.

2. Now suppose that you study your algorithm more closely and determine that the time complexity $T(n)$ is actually governed by the following inhomogeneous recurrence equality.

$$T(n+2) = 15 \cdot T(n+1) - 56 \cdot T(n) + 210$$

The initial conditions are now $T(0) = 10$ and $T(1) = 42$. Repeat the four items (a) through (d) of question 1, this time for this new inhomogeneous recurrence.

3. Now let

$$T(n+2) = 15 \cdot T(n+1) - 56 \cdot T(n) + 7^n \cdot 6$$

The initial conditions are now $T(0) = 4$ and $T(1) = 45$. Repeat the four items (a) through (d) of question 1, this time for this new inhomogeneous recurrence.

5DV022 Fall 2008, Problem Exercise 1, page 2

4. Finally, you are working with yet another, much improved algorithm, whose time complexity you have determined to be governed by the recurrence

$$T(n) = 5 \cdot T(n/3) - 6 \cdot T(n/9) + n^2.$$

The initial conditions are $T(1) = 5$ and $T(3) = 10$.

(a) Using a change of variable, obtain a linear recurrence relation.

(b) Give the characteristic polynomial of this linear recurrence.

(c) Solve for the roots of this polynomial.

(d) Give the solution to this linear recurrence.

(e) Translate this solution back to the original variable. For full credit, this answer must be in a simplified form.

(f) Give the best upper bound of complexity of the algorithm based upon this solution.

The identity $a^{\log_b(c)} = c^{\log_b(a)}$ may be useful in solving the above problem.

Note: For solving systems of linear equations, the free (as in speech as well as in beer) program Octave is highly recommended. It is installed on the departmental Linux systems, and it is very easy to use. There are many tutorials available on the Web.