CS 313	**Analysis of Algorithms**	Test 1
Name:	Summer 1999	Max Pts: 40

Important: This is an open book test. You can use any books, notes, or paper. Any calculations and rough work can be done on the back side of the test pages. You will loose five points for not writing your name.

1. [10 pt] Use mathematical induction to show that the solution of the recurrence

$$T_n = \begin{cases} 2 & \text{if } n = 2, \\ 2T_{\frac{n}{2}} + n & \text{if } n = 2^k,\ k > 1 \end{cases}$$

   is $T_n = n \lg n$.

2. [10 pt] Use iteration to solve the recurrence

$$T_n = T_{n-a} + T_a + n$$

   where $a \geq 1$ is a constant.

3. [10 pt] Find an asymptotic upper bound on the summation

$$\sum_{i=0}^{\lfloor \lg n \rfloor} \left\lceil \frac{n}{2^k} \right\rceil$$

4. [5 pt] Find a simple formula for

$$\sum_{k=1}^{n} (2k-1)$$

5. [5 pt] Use master method to determine the solution to the recurrence

$$T_n = T_{\frac{n}{2}} + \Theta(1)$$