Computer Science 3711 (Winter 2004):  
Assignment #2, Questions #2–5  
Due: 3:30 PM on Tuesday, February 10, 2004

2. (8 marks) For each of the situations described in the parts of this question, state the corresponding problem in terms of its inputs and outputs along the lines of the statement of the sorting problem on page 5 of the textbook.

a) (4 marks) You are organizing a bus tour for a group of tourists from Ottawa. You do not know the size of this group in advance, but you do know that many of the people in this group hate each other desperately. You will be told the pairs of people who hate each other and thus cannot be on the same bus during a tour. Given all this information, you want to know if it is possible to assign the group members to three buses such that no pair of people on a bus hate each other.

b) (4 marks) Two weeks ago, you gave your favorite nephew an amount of money to invest in savings bonds. On arriving at your nephew’s apartment, you are chagrined to find a mound of empty beer bottles, a stack of DVDs, and some savings bond certificates. Each DVD still has its cost-sticker attached and the purchase-price of each savings bond is written on the front of its certificate. During somewhat heated questioning, your nephew claims that he bought at least five savings bonds with the money you gave him and spent the rest on DVDs – that is, not one cent of what you gave him went to beer. You want to know if it is possible that your nephew is telling the truth.

3. (15 marks) Prove or disprove the following:

a) (5 marks) \( f(n) = 2^{n+1000d} \), where \( d \) is some integer constant greater than 0, is not \( O(2^n) \)

b) (5 marks) \( f(n) = n(n + \log n) \) is \( \Theta(n^2 \log n) \)

c) (5 marks) \( f(n) = n(n + 500)(n + 5) \) is not \( \Theta(n^3) \)

All proofs should give appropriate bounds on values of \( c \) and \( n_0 \).

4. (12 marks) Solve the following recurrences using the iterative method:

a) (6 marks)
\[
T(n) = \begin{cases} 
100 & n = 1 \\
T(n-1) + cn & n > 1 
\end{cases}
\]

b) (6 marks)
\[
T(n) = \begin{cases} 
0 & n \leq 1 \\
9T(n/3) + cn^2 & n > 1 
\end{cases}
\]
5. (15 marks) For each of the algorithms below, derive a worst-case time complexity function $T(n)$ and prove the tightest possible asymptotic upper bound on $T(n)$.

a) (6 marks)

sum = 0
for i = 1 to n do
    x = P1(n)
    tsum = x + 5
    for j = 1 to n * n do
        y = x / P2(n)
        tsum = tsum / y
    if (P3(n, tsum)
        tsum = tsum * x
    if (P4(n, tsum)
        tsum = tsum + tsum
return tsum

The running times of procedures P1(), P2(), P3(), and P4() are unknown at this time.

b) (9 marks)

REC-ALG(n)
if (n == 1)
    for i = 1 to n do
        write n;
else
    REC-ALG(n/3)
    FUNKY(n)
    REC-ALG(n/3)
    FUNKY1([square root of n])
    REC-ALG(n/3)

Note that procedures FUNKY(n) and FUNKY2(n) run in $O(\log_2 n)$ and $O(n^2)$ time, respectively.