Introduction To Algorithms
CS430

Fall 2015
HomeWork 2
Due 14th September

1. Consider lopsided trees with costs 3 and 1 on the left and right branches, respectively, of the search tree. Characterize the weights (costs) at the leaves of the optimal trees. Establish a recurrence for the maximum number of leaf nodes of cost w(No need to solve the recurrence). (20 pts)

2. (15 pts) (i) Set up time complexity for the following programs and solve the following recurrences:

   $L = (l_1, l_2 \dots l_n)$ is a list of size n
   SolveProb(L)
   {
   If $n > 1$\{
   SolveProb($L_1 = (l_1, l_2 \dots l_{n-1})$)
   SolveProb($L_2 = (l_2, l_3 \dots l_n)$)
   SolveProb($L_3 = (l_2, l_3, \dots l_{n-1})$) }
   }

   (ii) Solve the following recurrence:

   $$T(n) = 4T(n-1) + 2^n \ , n > 0$$

   with $T(0) = 1$.

   (iii) Solve

   $$T(n) = 8T(n-1) - 16T(n-2), \ n > 1$$

   with $T(0) = 2, T(1) = 10$

3. (5pts) Use the recursion tree method to solve

   $$T(n) = T(n-c) + T(c) + \sqrt{n}$$

   where c is a constant and $T(c) = c$

4. (10pts) Solve the recurrence

   $$T(n) = T(8n/9) + T(n/18) + f(n)$$

   when $f(n) = n$ and $f(n) = n^2$.

5. (10pts) Solve the recurrence

   $$T(n) = T(8n/9) + T(3n/18) + f(n)$$

   when $f(n) = \log n$ and $f(n) = n^2$.

6. (10pts) Solve the following recurrences where:

$$T(i) = 1;\ i = 1...10$$

(a) $T(n) = T(logn) + 3$

(b) $T(n) = T(logn) + 3n$