[redacted]	September 15, 2021
Problem Set 2	
[redacted]	*Due: 8 am on Thursday, September 23*

Problem 2-1 (Different Methods for Recurrences) 14+10 points

Consider the recurrence $T(n) = 2T\left(\frac{n}{2}+1\right) + n - 2$. Further, let $T(3) = 0$. Assume that n is of the appropriate form such that it is always an integer, even as we go down the formula.

(a) (8 points) Prove that $T(n) = \Theta((n-2)\log(n-2))$ using the "guess-then-verify" method. (**Hint**: Recall that if $f(n) = \Theta(g(n))$, then there exists constants $c_1, c_2 > 0$ such that $c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0 > 0$.)

(b) (**Extra Credit**)(6 points) Solve the same recurrence relation by the domain-range substitution. Namely, make several changes of variables until you get a basic recurrence of the form $R(k) = R(k-1) + f(k)$ for some function f, and then compute the answer from there. This answer will have to be exact and not an asymptotic answer. For full credit, you will have to each of the intermediate substitutions and calculations. (**Hint**: As a first step, you will let $n = c^m + d$ for some constants c, d. What is this choice of c, d?)

(c) (6 points) Consider the recurrence relation

$$T(n) = 2T\,(n-1) + n,\ T(0) = 1.$$

Solve by recursion tree method. Towards this end, you will first complete the empty empty fields at this table. You can then conclude that $T(n)$ is the sum of the values in the last column. You will then simplify the sum to give a tight asymptotic bound for $T(n)$, i.e., $T(n) = \Theta(c^n \cdot n^d \cdot (\log n)^r)$ for some constants $c > 1, d, r > 0$.

(**Extra Credit**)(3 points) You get an additional three points if you solve for the value of $T(n)$ exactly and correctly.

(**Hint**: You will find the following formula very useful: Let

$$S(a,d) = \sum_{i=1}^{d} i \cdot a^i = a + 2a^2 + 3a^3 + \ldots + d \cdot a^d = \frac{d \cdot a^{d+2} - a^{d+1}\cdot(d+1) + a}{(a-1)^2} = \Theta(d \cdot a^d)$$

)

Level	Size of Problem	No. of Problems	Non-Recursive Cost of One Problem	Total Cost
0				
1				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$d-1$				
d				

Here, d is the depth of the recursion tree. (**Hint**: To produce a closed form solution)

Problem 2-2 (Fun with Recurrences) 15+1 Points

(a) (5 points) Consider the following recurrence:

$$T(n,1) = 3n$$
$$T(1,m) = 3m$$
$$T(n,m) = 3n + T(n/3, m/3)$$

Solve for $T(n, n^2)$ to get a tight asymptotic bound. Assume that n is an exponent of 3 for simplicity.

Extra Credit (1 point): Keep track of the leading coefficient rather than just state $T(n) = \Theta(f(n))$. This means that express $T(n) = c \cdot f(n) + o(f(n))$ for an appropriate choice $c, f(n)$.

(b) (5 points) Consider the following recurrence

$$T(n) = T(0.01n) + T(0.99n) + cn$$

where $c > 0$ is a constant. Solve for $T(n)$ by the recursion tree method to derive an asymptotically tight solution. (**Hint**: Note that this tree is not going to be a balanced tree. What is the depth of the shortest branch? What is the depth of the largest branch? Use this to derive a Θ bound for $T(n)$.)

(c) (5 Points) Consider the following recurrence

$$T(n) = 9T(n/3) + \frac{n^2}{\log_3 n}.$$

Solve for $T(n)$ by domain-range substitution. Namely, make several changes of variables until you get a basic recurrence of the form $R(k) = R(k-1) + f(k)$ for some f, and then compute the answer from there. You may assume that n is a power of 3. (**Hint**: Begin by expressing n as a function of another variable m. Then change the variable T. You will change T to a function S and then to R. Note: The two changes can be combined in one step too.)

Problem 2-3 (Expected Running Time) 7+5 points

Consider the pseudocode for the following randomized algorithm:

BLA(n)
  if $n \leq 5$ **then return** 1
  else
    Assign x value of 0 with probability 1/4 and 1 with probability 3/4
    if $x = 1$ **then return** BLA(n)
    else return BLA($n/3$)

(a) (5 points) Let $T(n)$ denote the expected running time of BLA. Derive a recurrence equation for $T(n)$. Solve your recurrence relation using Master Theorem to obtain a asymptotically tight bound.

(b) (2 points) What is the functionality of this algorithm, i.e., what is the expected value returned by this algorithm? Justify your answer briefly.

Consider the pseudocode for the following randomized algorithm:

FOO(n)
  if $n \leq 1$ **then return** 5
  else
    for $i = 1$ **to** n **do**
      continue
    Assign x value of 0 with probability 1/4 and 1 with probability 3/4
    if $x = 1$ **then return** FOO(n)
    else return BLA(FOO($n/3$))

(c) (**Extra Credit**)(5 points) Let $S(n)$ denote the expected running time of FOO. Derive a recurrence equation for $S(n)$. Solve your recurrence relation by the Recursion Tree Method, by completing the table as shown below: