1. (50 points) Assume that $T(1) \in \Theta(1)$. Solve the following recurrence functions using the master method. If the master method cannot be applied, state the reason, and give an upper bound (big-Oh) as tight as you can. Justify your answer.

   a. $T(n) = 8T(n/2) + n^2$;
   b. $T(n) = T(3n/5) + n$;
   c. $T(n) = 9T(n/3) + n^2$;
   d. $T(n) = 16T(n/4) + n \log n$;
   e. $T(n) = 2T(n/4) + \log^2 n$;
   f. $T(n) = 3T(n/3) + \log n$;
   g. $T(n) = 4T(n/4) + n \log n$;
   h. $T(n) = 2T(n/4) + \sqrt{n}$;
   i. $T(n) = 3T(n/3) + (n + \log n)$;
   j. $T(n) = 2T(n/2) + n/\log n$;

2. (Optional, 10 points) Assume that $T(1) \in \Theta(1)$. Solve the following recurrence functions using the master method and change of variables.

   a. $T(n) = 3T(n - 2) + n$;
   b. $T(n) = T(n - 2) + n^3$;

3. (20 points) Assume that $T(1) \in \Theta(1)$ and $T(n) = T(3n/4) + T(n/2) + n^2$. Prove $T(n) \in \Theta(n^2)$ using the substitution method.

4. (Optional, 10 points) Assume that $T(1) \in \Theta(1)$ and $T(n) = 3T(n - 1) + n^2$. Prove $T(n) \in O(3^n)$ using the substitution method.

Continued on the back →
5. (30 points) Analysis of recursive algorithm. Consider the pseudocode of the following two algorithms. The input $A$ is an array of size $n$. In Alg1, $A$ is divided into two subarrays, and the algorithm is recursively applied to only one subarray. In Alg2, $A$ is divided into five subarrays, and the algorithm is recursively applied to two or three of them, depending on values in $A$. Assume that parameter passing takes constant time.

```
Alg1 (A[1..n])
  if (n <= 1) return;
  mid = floor (n/2);
  x = rand(); // 0 < x < 1
  if (x <= 0.5)
    Alg1 (A[1..mid]);
  else
    Alg1 (A[mid+1..n]);
end

Alg2 (A[1..n])
  if (n <= 1) return;
  s = floor(n/5);
  x = rand(); // 0 < x < 1
  if (x <= 0.5)
    Alg2 (A[1..s]);
    Alg2 (A[s+1..2s]);
  else
    Alg2 (A[2s+1..3s]);
    Alg2 (A[3s+1..4s]);
    Alg2 (A[4s+1..n]);
end
```

a. What is the worst-case running time of Alg1? What about best-case and average case?

b. What is the worst-case running time of Alg2? What about best-case and average case?

c. We have assumed that it parameter passing takes constant time. Now let’s say it actually takes $f(n) = \Theta(n)$ time to pass an array of size $n$. Re-do (a) and (b).

6. Bonus (5 points) What do you think about the difficulty level of homework assignments (harder than expected? just all right? easy?) What is the most difficult part? Do you have any comments/suggestions about the lecture, recitation, and assignment?