Assignment 2 (10 points for each question)

1. (10 points) Show the solution of T(n) = T(n-1) + $\lg(n^2)$

2. (10 points each question) Use the recursion tree method to determine the asymptotic upper bounds for the following Recurrences

   a. T(n) = 4T($n-1$) + $\sqrt{n}lgn$
   b. T(n) = T(0.2n) + T(0.8n) + 3n

3. (4 points each question) Use the Master Theory to solve the following recurrences

   a. T(n) = 3T(n/27) + 1
   b. T(n) = 7T(n/8) + lgn
   c. T(n) = 2T(n/4) + n
   d. T(n) = 2T(n/4) + n^2
   e. T(n) = 2T(n/4) +$\sqrt{n}$ lg*n*

4. (Textbook 4.5-5 page 97) Consider the regularity condition $af(n/b) \leq cf(n)$ for some constant $c < 1$, which is part of case 3 of the master theorem. Give an example of constants $a \geq 1$ and $b > 1$ and a function f(*n*) that satisfies all the conditions in case 3 of the master theorem except the regularity condition.

5. Show that in any subtree of a max-heap, the root of the subtree contains the largest value occurring anywhere in that subtree.

6. Illustrate the operation of MAX-HEAPIFY (A, 1) on the array A = {27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0}.

7. Show that O(n) is the asymptotic upper bound of the number of swap operations that are performed by Build-MAX-Heap function to build a max heap on an array A of *n* elements.

8. (Textbook 6.4-1 page 160) Illustrate the operation of HEAPSORT on the array A = {5, 13, 2, 25, 7, 17, 20, 8, 4}.

9. For HEAPSORT codes below

   Heapsort(A)
   {

```
        Build-MAX-Heap(A);
        for (i = A.length downto 2)
        {
                Swap(A[1], A[i]);
                A.heap_size= A.heap_size - 1;
                MAX-Heapify(A, 1);
        }
}
```

(a) (3 points) What is the number of required swap operations when heapsort the array A = {5, 13, 2, 25, 7, 17, 20, 8, 4}? Explain your reason.

(b) (3 points) If we replace MAX-Heapify(A, 1) with Build-MAX-Heap(A), what is the number of required swap operations when heapsort the array A? Explain your reason.

(c) (4 points) Does the asymptotic upper bound of Heapsort increase from O(nlgn) to O(n^2)? Why? (Hint: compare the number of swap operations before and after the change for the worst case).