# Homework 4
# Com S 311
# Due: Oct 26, 11:59PM

Late Submission Due: Oct 27, 11:59PM (25% penalty)

**Outcomes.**

- Understand Divide and Conquer

**Please see guidelines for submission instructions.**

If you are using divide and conquer to solve the problems, then your solution must clearly and specify describe each of the following steps:

- Dividing the problem into sub problems
- Solving sub problems
- Merging solutions of sub problems.

Then, you must derive a recurrence relation for the run-time of the algorithm, and solve the recurrence relation. In this PA, you may assume that $n$ is a power of 2 (if it helps).

There are 5 problems and each problem is worth 50 points.

1. Consider the following array.
$$10, 8, 13, 9, 12, 5, 52, 2, 59$$
   (a) Construct the max heap for the above array. Draw the resultant heap as a tree.
   (b) Show the steps necessary to delete 59 from the heap. Draw the resultant heap (as a tree)

2. Solve the following recurrences. You can not use master theorem to solve them. You must show the steps in your derivation.
   (a) $T(n) = T(n/3) + T(2n/3) + cn$.
   (b) $T(n) = T(n/5) + cn$.
   (c) $T(n) = 2T(n/2) + n^{log_5(7)}$

3. Suppose you are given an array of $n$ integers. Assume that $n \geq 2$. Your goal is to find largest and second largest elements of the array. Consider the following algorithm:

```
If A[1] < A[2],
   largest = A[2], Second = A[1]
 else
    largest = A[1], Second = A[2];

 For i in range 3 to n {
   If A[i] > largest  {
        second = largest;
        largest = A[i];
    }
   else
      if A[i] > second
         second  = A[i]
 }
```

- In worst-case, how many comparisons are made *among the elements of A*? Give the exact number of comparisons in worst-case, not asymptotic bound. I.e, Your answer must be an expression such as $2n$ or $3n - 4$ instead of $O(n)$.
- Design an algorithm that reduces the number of comparisons made among the elements of $A$. Derive the number of comparisons. Your grade depends on the number of comparisons.

4. An array of integers $[a_1, a_2, \cdots, a_n]$ is *k-sorted* if the following condition holds: For every $i$, $1 \leq i \leq n - k$, $a[i] \leq a[i + k]$. Design an algorithm that gets a $k$-sorted array as input and sorts it. Describe your algorithm and derive the run time. Express runtime in terms of $n$ and $k$. Your grade partly depends on the efficiency of your algorithm.

5. Let $A = [a_1, a_2, \cdots, a_n]$ be an array of integers. Given $1 \leq i < j \leq n$,

$$S_{ij} = \sum_{k=i}^{j} a_k$$

Give an algorithm to compute $\max_{i,j} S_{ij}$, analyze the run-time. Your grade depends on the run-time of your algorithm.

**GUIDE LINES:**

- It is important to know whether your really know! For each problem, if you write the statement "I do not know how to solve this problem" (and nothing else), you will receive 20% credit for that problem. If you do write a solution, then your grade could be anywhere between 0% to 100%. To receive this 20% credit, you must explicitly state that you do not know how to solve the problem.
- You must work on the homework problems on your own. You should write the final solutions alone, without consulting any one. Your writing should demonstrate that you understand the proofs completely.

- When proofs are required, you should make them both clear and rigorous. Do not hand waive.
- If you hand writing is not legible, then your homework will not be graded.
- Any concerns about grading should be made within one week of returning the homework.
- **Please submit your HW via blackboard. If you type your solutions, then please submit pdf version. If you hand-write your solutions, then please scan your solutions and submit a pdf version. Please make sure that the quality of the scan is good, and your hand writing is legible. Name your file must be** $YourNetID$**-**$HW4.pdf$**. For example, if your net id is** `bondj`**, then the file name should be** $bondj$**-**$HW4.pdf$**. HW's submitted in incorrect format (non pdf, incorrect file name etc) will incur a penalty of 30%**