Homework #2

You should try to solve these problems by yourself. I recommend that you start early and get help in office hours if needed. If you find it helpful to discuss problems with other students, go for it. **The goal is to be ready for the in class quiz that will cover the same or similar problems.**

Problem 1: Master Method

Use the master method to give a tight asymptotic bound for each of the following recurrences.

1. $T(n) = 8T(n/2) + \Theta(n^3)$
2. $T(n) = 3T(n/2) + \Theta(n)$
3. $T(n) = 3T(n/2) + \Theta(n^2)$
4. $T(n) = 16T(n/2) + \Theta(n^3)$
5. $T(n) = T(9n/10) + \Theta(n)$

Problem 2: Recurrence Relations

Calculate the time complexity of the below divide-and-conquer algorithm via a recurrence relation. Assume addition is an $\mathcal{O}(1)$.

```
getResult(A[0...n - 1]){
   if (n = 0) { return 0 }
   else if  (n = 1)  { return A[0] }
   i =  n/4
   R1 = getResult(A[0...2i - 1])
   R2 = getResult(A[i...3i - 1])
   R3 = getResult(A[2i...n - 1])
   return R1 + R2 + R3
}
```

(a) Write a recurrence relation for `getResult`.

(b) Solve the recurrence relation to find a big $\mathcal{O}$ expression for the number of `getResult` calls as a function of n.

Problem 3: Maximum in a Shifted Array

Suppose you are given an array A of n sorted numbers that has been *circularly shifted* to the right by k positions. For example $\{35, 42, 5, 15, 27, 29\}$ is a sorted array that has been circularly shifted $k = 2$ positions, while $\{27, 29, 35, 42, 4, 15\}$ has been shifted $k = 4$ positions. Give an $O(\log n)$ algorithm to find the largest number in A. You may assume the elements of A are distinct. Write the recurrence for your algorithm and show that its recurrence solves to $O(\log n)$ (e.g., using the Master Method, a recursion tree, or an inductive proof).

Problem 4: Finding the Extra Element

Suppose you are given two sorted arrays A and B in which there is exactly one difference between the two: A contains one extra element that is not in B. We want to find the index of this extra element. For example, if $A = \{2, 4, 6, 8, 9, 10\}$ and $B = \{2, 4, 6, 8, 10\}$, then the extra element is 9. Write the recurrence relation for your algorithm and show that its recurrence solves to $O(\log n)$ (e.g., using the Master Method, a recursion tree, or the iteration method).