

# CS344: Design and Analysis of Computer Algorithms

## Homework 3

**Group Members: Stephen Kuo, Derek Mui**

2.4) Suppose you are choosing between the following three algorithms:

- Algorithm *A* solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.
- Algorithm *B* solves problems of size  $n$  by recursively solving two subproblems of size  $n - 1$  and then combining the solutions in constant time.
- Algorithm *C* solves problems of size  $n$  by dividing them into nine subproblems of size  $n/3$ , recursively solving each subproblem, and then combining the solutions in  $O(n^2)$  time.

**Answer:**

- $T(n) = 5 * T(n/2) + cn$ .  
Applying master theorem  $a = 2, b = 5, f(n) = c n, \text{degree}(f(n)) = 1$   
Since  $\log_2 5 > 1, T(n) = O(n^{\log_a b}) = O(n^{\log_2 5})$
- $T(n) = 2T(n - 1) + c. \therefore T(n) = O(2^n)$
- $T(n) = 9T(\frac{n}{3}) + cn^2$ .  
Applying master theorem  $a = 3, b = 9, f(n) = cn^2, \text{degree}(f(n)) = 2$   
Since  $\log_3 9 = 2, T(n) = O(n^2 \log n)$

Time complexity of the third algorithm is the best.  $\therefore$  choose algorithm C

2.5) Solve the following recurrence relations and give a  $\Theta$  bound for each of them.

**Answer:**

2.14) You are given an array of  $n$  elements, and you notice that some of the elements are duplicates; that is, they appear more than once in the array. Show how to remove all duplicates from the array in time  $O(n \log n)$

**Answer:** Simply sort the elements of the array using merge sort in  $O(n \log n)$  time and then remove the duplicate elements by traversing the sorted array.

2.25) ?

**Answer:**

2.28) ?

**Answer:**

3.5) ?

**Answer:**

3.31) ?

**Answer:**

3.43) ?

**Answer:**