

# TCSS 343 - Week 3

Jake McKenzie

August 5, 2018

## Divide and Conquer

“Common misconception that *fun* is relaxing.  
If it is, youre not doing it right.”

...

Iain Banks

“Writing is nature’s way of letting you know how sloppy your thinking is.”

...

Guindon

“Do those things that incline you toward the big questions, and avoid the  
things that would reduce you and make you trivial.”

...

George Saunders

0. Use the master method to solve the following recurrences. If the master method does not apply indicate that it does not.

I)  $T(n) = 7T(\frac{n}{7}) + O(n)$

II)  $T(n) = 5T(\frac{n}{3}) + O(n)$

III)  $T(n) = 3T(\frac{2}{n}) + O(1)$

IV)  $T(n) = 16T(\frac{n}{4}) + O(n^3)$

V)  $T(n) = 4T(\frac{n}{9}) + O(n \log n)$

VI)  $T(n) = T(\frac{n}{9}) + O(\sqrt{n})$

VII)  $T(n) = 7T(\frac{n}{2}) + O(n^2)$

VIII)  $T(n) = 4T(\frac{n}{2}) + O(16^{\log n})$

IX)  $T(n) = 2T(n) + O(n^3)$

X)  $T(n) = 2T(\frac{n}{3}) + O(n^{-\log n})$

XI)  $T(n) = 2T(\frac{n}{2}) + O(n^{0.51})$

1. Consider the problem of searching an element  $x$  in an array  $\text{arr}[]$  of size  $n$ . The problem can be solved in  $O(\log n)$  time if:

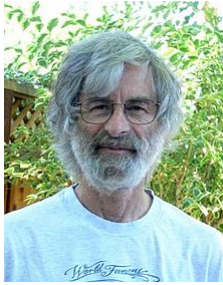
- I) Array is sorted
- II) Array is sorted and rotated by  $k$ .  $k$  is given to you and  $k \leq n$
- III) Array is sorted and rotated by  $k$ .  $k$  is NOT given to you and  $k \leq n$
- IV) Array is not sorted

- a) I
- b) I and II
- c) I,II and III
- d) I,II,III and IV

2. What is the probability that a number is composite numbers in first 50 natural numbers inclusive? (hint: 1 is neither composite nor prime and composite). (I've seen this question in pools of interview questions for algorithms)

- a) 0.58
- b) 0.62
- c) 0.68
- d) 0.72

3. Solve the following recurrence:  $T(n) = T(\frac{6n}{7}) + T(\frac{n}{7}) + O(n)$ . The master method cannot be used on this one. Use the tree method or repeated substitution to find the runtime.



Thinking doesn't guarantee that we won't make mistakes. But not thinking guarantees that we will. ~ Leslie Lamport (turing award winner, both devised important algorithms and ways of formal modeling)

4. **Preferably by using your notes, write down a formal specification for quicksort.** Formal specifications are important, they give us the language to describe subtle problems that you can't get from simply writing code. When students hear "specifications" they typically panic. You have to learn and use these funny symbols and such, but if you don't get it right that's okay. The world isn't going to come crashing down if you don't do get it perfectly right but please try.



Writing formal specifications won't catch coding errors or bugs, but it will catch algorithm errors. ~ Leslie Lamport (the guy from the previous page)

5. You typically write specifications while thinking about the problem, before you write code. I'm assuming you probably wrote a recursive specification for quicksort on the previous page. If you didn't, great you're done you can move on. **If you did write a recursive specification for quicksort, please now attempt to write a non-recursive specification for quicksort.** This is more challenging but please, please try to do so. It requires more thought but thinking is a really good idea, don't trust anyone who tells you otherwise. Afterward come up with one reason why you think it's a good idea to write a spec.