

CS 624: Analysis of Algorithms

Fall 2019 Assignment 1

Due: Thursday Sep. 19, 2019

1. Exercise C.1-15 (page 1189) Hint: use a generating function and differentiate.
2. Decide whether each of the following statements is true or false, and prove that your conclusion is correct.
 - $2^{n+1} = O(2^n)$
 - $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$
3. Prove that $\log_a x = O(\log_b x)$ for any $a > 0$ and $b > 0$.
4. Prove that if $f = O(g)$ and $g = O(h)$ then $f = O(h)$.
5. Problem 4-1 (a, b, c, f, g) (page 107).
6. Problem 4.2 in Lecture notes 1 (page 7).
7. Problem 4.1 in Lecture notes 2 (page 12).
8. Exercise 6.5-8 (page 166). Note that the problem asks you to give an algorithm that runs in $O(\log n)$ time. So you not only have to give the algorithm, you also have to show that it really does run in $O(\log n)$ time.
9. Exercise 6.5-9 (page 166). Another hint: Think of the merging part of MergeSort and extend it to multiple lists. Remember that the lists are not necessarily the same size.
10. Exercise 6.1 in the Lecture 3 handout (on page 12 of the handout).
11. Problem 7-4 (page 188). This requires careful explanation. It's the kind of reasoning that is very important in software engineering.
12. Prove that if a binary tree has depth n , then it has at most 2^n leaves. Please be careful about this. The result is not "obvious" and you certainly can't assume that the tree is "all filled out", or that a tree that is "all filled out" has the maximum possible number of leaves. That's in fact what you are trying to prove, so you can't assume it is true. A good way to prove this result is by induction:
 - (a) The inductive hypothesis should be a sequence of statements, one for each possible depth of the tree. For instance,
 - If a binary tree has depth 5, then it has ...
 - ...Write down a few of these statements, each one completely (i.e., without "...").

- (b) Now finish writing the proof by induction, using these statements as the inductive hypothesis. The proof is not hard, but you really have to be careful. And let me just say it once again: you can not assume that any of the trees you are considering is "filled out completely". If you do, the proof is automatically wrong, and I won't even read it.