CS 624: Analysis of Algorithmns

Assignment 4

Due: Monday, February 22, 2021

Note: Problems 1–7 are actually quite fundamental. They cover material that comes up again and again in discrete mathematics and is used in practice in computer science. So really try to get used to this kind of reasoning—you will find it extremely useful.

You will discover that many of the exercises are proved using the results of previous ones. So you really have to do them in order!

> Also I'd like to make a request, and this really applies to everything you hand in. It's OK to write short snippets of pseudo-code—5–7 lines or so. But anything longer than that becomes really hard to read. In general, I think it is far better to simply describe any algorithm you are presenting in ordinary language. If what you have written is clear, then I assure you that I will be able to turn it into code, if I really have to. (And frankly, I never have to.)
>
> It's true that some of the pseudo-code in the lectures is longer than that. And we actually spend a fair amount of time in class going over it. It's not easy to just look at. I think that the algorithms I ask you to produce in the assignments are easy to describe without using pseudo-code.
>
> From time to time I have had students turn in pages of pseudocode that is almost impossible for me to understand. Please be nice to me! Thanks.

1. Exercise 1.1 in the Lecture 7 handout.
2. Exercise 1.2 in the Lecture 7 handout.
3. Exercise 1.3 in the Lecture 7 handout.
4. Exercise 1.4 in the Lecture 7 handout.
5. Exercise 1.5 in the Lecture 7 handout.
6. Exercise 1.6 in the Lecture 7 handout.
7. Exercise 4.1 in the Lecture 7 handout.

   Please be careful. This is an exercise about binary search trees, *not* about algorithms used to construct those trees. So all you can use in doing this problem is the definition of a binary search tree. If you write something like, "this can't happen because the algorithm would have placed this element somewhere else", then your reasoning can't possibly be correct.
8. Prove that an inorder traversal of a binary search tree vists the nodes in increasing order of their keys. (Hint: use induction. The inductive hypothesis could be this: "For each binary search tree of height $\leq n$ (where n is some fixed number), the inorder traversal of that tree visits the nodes in increasing order of their keys."

   Show that the inductive hypothesis is true for $n = 1$. Then show that if it is true for some n, it can be proved to be true for $n + 1$.