CS 624: Analysis of Algorithms

Assignment 1

Due: Monday, February 1, 2021

1. Let $\{f_n : n = 0, 1, \dots\}$ be the Fibonacci sequence (where by convention $f_0 = 0$ and $f_1 = 1$).

   (a) Prove that

   $$\sum_{n=1}^{\infty} \frac{n f_n}{2^{n-1}} = 20$$

   Do this by using a generating function as shown in the last section of the Lecture 2 notes, and differentiating.

   (b) Show why (in the same way as you proved the first part of this problem) you might think that

   $$\sum_{n=1}^{\infty} n f_n = 2$$

   Then show why this could not possibly be true[1].

2. A professor in this department was once asked by a friend who was an economics professor elsewhere[2] if there was a simple form for the sum of the binomial coefficients of odd index in a row of Pascal's triangle. That is, the question was—given any integer $n \geq 0$—to evaluate the sum

   $$\binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \cdots$$

   where the last term is $\binom{n}{m}$ where

   $$m = \begin{cases} n & \text{if } n \text{ is odd} \\ n-1 & \text{if } n \text{ is even} \end{cases}$$

[1] And I don't mean something like, "It seems implausible to me." or "I don't believe it.". It has to be something—a *reason*—that you could explain to anyone else, anywhere in the world, and that would cause them to be convinced without a doubt. It doesn't have to be a long explanation. In fact, it could be very short. And it doesn't have to use fancy mathematical notation. Simple is always good! But whatever it is, it has to be absolutely convincing. That's what we mean—and that's *all* we mean—by a proof.

[2] This is a true story.

This problem is quite easy (as the economics professor guessed it might be), but for people who are not used to this sort of thing it can seem pretty remarkable.

Here is a simple way to see what to do:

(a) Think of $(1+x)^n$ as the sum of a (finite) generating function. Write out what it is. (We did this, or something very similar to this, in class.)

(b) Do the same for $(1-x)^n$.

(c) Then see what $(1+x)^n + (1-x)^n$ would be.

(d) And while you're at it, see what $(1+x)^n - (1-x)^n$ would be.

(e) At that point you should be just about done. (Well, I suppose at some point toward the end of this, you'll want to set $x = 1$.) Use what you have found to get the answer to the question the economics professor asked.

3. Decide whether each of the following statements is true or false, and prove that your conclusion is correct.

   - $n^2 = O(2^n)$
   - $2^{n+1} = O(2^n)$
   - $2^{2n} = O(2^n)$
   - $f(n) = O\big(g(n)\big)$ implies $2^{f(n)} = O\big(2^{g(n)}\big)$

4. Prove that $\log_a x = O(\log_b x)$ for any $a > 1$ and $b > 1$.

5. Suppose $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ is a polynomial.

   (a) If $0 \le j < n$, prove that $x^j = O(x^n)$.

   (b) Suppose you know that $a_n > 0$. Prove carefully that $p(x) = O(x^n)$. (OK, I'll give you a hint. Probably the easiest way to do this is to use Lemma 3.1 from Lecture 2.)

   (c) In fact, under the same assumption (i.e., that $a_n > 0$), you can also show that $p(x) = \Omega(x^n)$. Prove this as well. (Note that together with the previous item, this shows that $p(x) = \Theta(x^n)$.)

   (d) Suppose you know that $p(x) = O(x^d)$ for some number d such that $d < n$. Prove that $a_n = 0$.

6. Prove that if $f = O(g)$ and $g = O(h)$ then $f = O(h)$.

7. Problem 4-1 (a, b, c, f, g) (page 107).

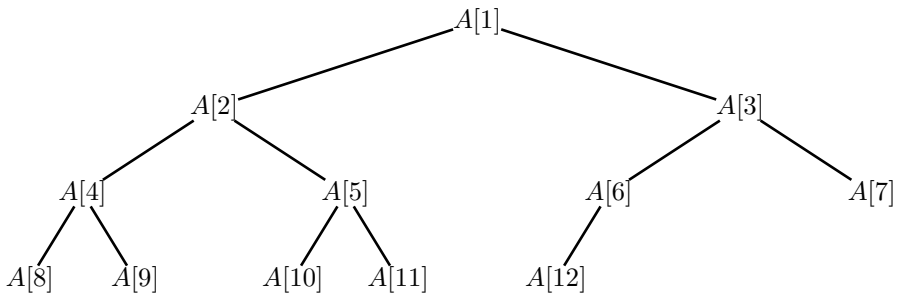
8. Exercise 4.1 in the Lecture 2 handout (page 13 in the handout).

9. Suppose we have n points in the plane, each one specified by its two coordinates, and suppose we want to write a program that will find a pair of points that are closest together. Here is an algorithm that does this; it is what is called a *brute-force search*: Make a list of all the pairs of points. Walk down this list, and for each pair, figure out the distance between the two points. Keep track of the smallest distance you have found so far, and the two points that gave you that distance. When you have finished, you will have found a pair of points with the smallest distance between them.

   (a) Could there be more than one such pair? (If so, you have to give an example. If not, you have to prove that would be impossible.)

(b) Show that the running time of this algorithm is $\Theta(n^2)$ (where n is the number of points)[3].

10. Suppose we have a 1-based array $A[1..n]$ which holds the elements of a binary tree, like this:



And we further assume that each row is filled in from the left, and the last row may not be complete, but it too is filled in from the left.

The point of this exercise is to prove that in this tree (well actually, in *any* tree of this shape), the children of $A[n]$ (if they exist) are $A[2n]$ and $A[2n+1]$. I want you to do this as follows:

Let us say that the root node of the tree is at "level 0". Its two children are at "level 1". And so on. You should be able to see that in any row (say at level k) which is completely filled in (in other words, any row except possibly the last one), the first element is $A[2^k]$ and the last one is $A[2^{k+1}-1]$. That's one of the things we will prove below, but you should make sure that it makes sense to you first. **Don't go on until it does make sense, and if it doesn't, then send me email right away.**

I want to show you how you can actually prove this is the case:

What we are going to assume: Let $n \geq 0$ be a **fixed** integer. Suppose that

(a) Level n is all filled out.

(b) The first element in level n is $A[2^n]$.

(c) The last element in level n is $A[2^{n+1}-1]$.

What you have to show: Show (based only on these assumptions) that

(a) The children of element $A[k]$ in level n (if they exist) are $A[2k]$ and $A[2k+1]$.

(b) And based on that,

i. The first element in level $n+1$ (if it exists) is $A[2^{n+1}]$ and

ii. If level $n+1$ is all filled out, then the the last element in level $n+1$ is $A[2^{n+2}-1]$.

You have to be careful when doing this. You have to start at the left end of the new row and "walk across the row to the right". This amounts to a proof by induction. See if you can get it right.

And **please** remember that the value of n does not change at any point in this proof.

[3]It may seem that you couldn't do better; but actually there *are* more efficient algorithms.