

Homework 12 Essay

This semester I have grown as a computer scientist in three major ways: I learned a lot of new concepts, improved my communication skills, and strengthened my analytical thought process.

Computer Science Concepts

Taking Discrete Structures taught me a lot of different concepts. It was interesting to learn proof techniques such as Proof by Contradiction and Proof by Induction. These concepts turned out to be very powerful when thought about in the right way. In the past, I have wondered how mathematicians have proved theorems for infinite sets of elements and I enjoyed learning the basics of elegant proof techniques.

Another concept that I found interesting was Big-O notation. I had encountered Big-O notation before this class, but I did not understand how it connected to recurrence relations and time complexity algorithms. In previous classes, Big-O notation has been defined rather ambiguously. I have mostly just memorized which types of search/sort algorithms connect to different time complexity measurements. However, Discrete Structures provided solid definitions for Big-O and other types of time complexity notations for defining the efficiency of different algorithms.

I also learned a lot about graphs and trees in this course. Before this class, I had no concept for what graphs or trees were or what they were useful for. I enjoyed learning about how to represent real life problems in terms of nodes and edges and then applying theorems that we learned to find solutions for issues that come across in every day life. In dealing with graphs and trees, I was especially interested in the 4 Color Theorem book. Diving deep into the formation of one, complex proof was fascinating. It was also extremely helpful to see a well thought out proof that had been edited specifically to communicate a complex idea in a rather simple manner. It provided a good example for me to base my proofs off while writing my homeworks.

I was also interested in combinatorics and probability. Having taken a statistics course in high school, I had an elementary grounding in these issues, but nothing that felt extremely useful. However, the content in discrete structures seemed to up the game. I felt challenged and interested in the outcome of the problems that we discussed.

Learning all these concepts helps me become a better computer scientist because of how directly they relate to the kinds of problems that I will interact with in the real world. Issues such as deciding which algorithm is the most efficient and inductively proving that a loop will terminate will be invaluable in the workforce.

Communication

Another aspect of my abilities in Computer Science that changed as a result of taking Discrete Structures was my communication skills. I have always been rather sloppy in my mathematic homework, ie not bothering to name all my variables or leaving out why things are true, but taking this class has radically changed that. I now find myself fully expressing my thoughts and stating why the work I did was true. This can be seen in an example from past homeworks.

In Appendix A, I give the statement that must be proven, and while I do show some steps, I do not use complete sentences or fully communicate my thought process. Someone who is familiar with mathematics could take some time and follow my logic, but it would be far easier for me to write out my thinking. Doing this, allows for easier communication and gives me practice fully explaining myself, which will help me in the workforce. A good example of how far my work has come is shown in Appendix B. In the problem in Appendix B, I define my variables and explain the logic of how arrived at the numbers that I used.

Another way that I improved my communication skills in Discrete Structures was using a LaTeX editor. I did not have much experience with LaTeX and learning how to write tables and clean mathematical notation in a document was not only satisfying, but is extremely useful in this paperless world.

Analytical Thought

In my opinion, the most valuable skill that I honed in Discrete Structures was using analytical thinking. I feel that Discrete Structures has given me a lot of tools to tackle interesting and tough statement regarding issues that relate to mathematics.

A good example of this was learning Proof by Induction. Induction is somewhat of a backwards concept that I never would have even dreamed of outside this class. Learning the steps and the way you have to think to apply induction provided a good puzzle for my brain that has made my mind much more nimble. At the beginning, every induction problem seemed impossible. But with more practice, I was able to see patterns that continued to show up in the examples that we did and I began to get a grasp on the issue. Towards the end of our work on induction, I felt confident applying it to the loop-guard problem that we did in recitation one day. There were some issues with my application of it on the loop-guard, but I did not need any help because of the grounding in analytical thinking that Discrete Structures gave me. Proof by Contradiction was another concept that challenged my thought process. It was awkward, at first, to just assume that something was true, but, again, more practice provided me with more comfortability and confidence.

Conclusion

Overall, Discrete Structures was an interesting class that provided me with a wide array of proof techniques and challenged the way I think about mathematics, all while improving my technical communication skills.

A Homework 2

Question: Prove $\binom{n+3}{n+1} = \frac{(n+3) * (n+1)}{2}$

Proof:

$$\begin{aligned} \binom{n+3}{n+1} &= \frac{(n+3) * (n+1)}{2} \\ \frac{(n+3)!}{(n+1)! * ((n+3) - (n+1))!} &= \\ \frac{(n+3) * (n+2) * \cancel{(n+1)!}}{\cancel{(n+1)!} * 2!} &= \\ \frac{(n+3) * (n+1)}{2} &= \frac{(n+3) * (n+1)}{2} \end{aligned}$$

B Homework 10

Section 9.2, Problem 17

- Let n_i be the number integers from 1000 to 9999. The first digit has nine options (integers 1 through 9), and the remaining three have ten options (integers 0 through 9). $n_i = 9 * 10 * 10 * 10 = 9000$
- Let n_o be the number of integers from 1000 to 9999 that are odd. The first digit has nine options (integers 1 through 9) and the middle two have ten options (integers 0 through 9). The final digit must be in the set $A = \{1, 3, 5, 7, 9\}$ to make the integer even, the size of A is 5, therefore, the last digit has 5 options. $n_o = 9 * 10 * 10 * 5 = 4500$
- Let n_d be the number of integers from 1000 to 9999 that are distinct. The first digit has nine options (integers 1 through 9) and the second digit also has nine options (integers 0 through 9 excluding the first choice). Each subsequent digit has one less option than the digit before it. $n_d = 9 * 9 * (9 - 1) * (9 - 2) = 9 * 9 * 8 * 7 = 4536$
- Let n_b be the number of integers from 1000 to 9999 that are both odd and distinct. To be odd, the final digit of a number must be odd. Let $A = \{1, 3, 5, 7, 9\}$ be the set of odd digits. We will now find the amount of distinct 4-digit numbers that can be created given we have selected a final digit. Given an arbitrary final digit in the set A , there are 8 options for the first digit (integers 1-9 excluding final digit), the second digit also has 8 options (integers 0-9 excluding the first digit and the final digit), and the third digit has 7 options (integers 0-9 excluding the first and second digits as well as the final digit). Because an arbitrary final digit was chosen, this product can be multiplied by the magnitude of A to find n_b . $n_b = 8 * 8 * 7 * |A| = 8 * 8 * 7 * 5 = 2240$.
- $P(\text{A random 4-digit integer has distinct digits}) = \frac{\text{number of 4-digit integers with distinct digits}}{\text{number of 4 digit integers}} = \frac{4536}{9000} = 0.504$
 $P(\text{A random 4-digit integer has distinct digits and is odd}) = \frac{\text{number of odd 4-digit integers with distinct digits}}{\text{number of 4 digit integers}} = \frac{2240}{9000} = 0.2488$