Discrete Structures, Homework n+1=9

Patrick O'Connor

Due: 30 April 2021

This homework assignment should be submitted as a single PDF file to Gradescope.

Write a two-page paper describing to me how you have grown as a student, computer scientist, mathematician, engineer, or a researcher in this class, and more generally, in this semester. To support your argument, you should include your homework or writing samples (or excerpts from them) in an appendix as evidence (and reference them!)

If you do not feel that you've grown, explain why.

Remember, style counts. Use complete sentences.

This HW will be graded on the following scale:

- No submission (0%)
- Low pass (70%)
- Pass (90%)
- High Pass (100%)

Throughout this semester I have grown as a student, computer scientist and mathematician. I have grown as a student by learning new technology and techniques for proving and understanding topics. Along with this I have grown as a computer scientist by learning more about graphs and big O notation. Lastly I have grown as a mathematician in that I have increased my knowledge of different areas of mathematics that are not taught in the average mathematics course.

Going through the above list as a student I am happy to have learned of this new technology called latex. Since starting this course I have truly learned to appreciatte the abilities of latex and what it can achieve. I have started to use Latex to typeset homework assignments for other classes, in which I have seen an increase in the quality of my final project work. Along with this I have since created a new resume using the libraries that are provied and created a quite impressive looking resume. Along with this I have since increased my problem solving skills by introducing new techniques of attacking a problem such as understanding proofs and some what understanding the creation of reducible graphs as they did in the Four Color Suffice proof presented in the book.

Along with this as a computer scientist I have increased my knowledge of graphs and Big O/function complexity. Throughout this semester I have solidfy my knowledge of many of the concepts that were taught to me throughtout csci-232 such as trees and graphs in general. When I initially learned about these mathematical structures much of my understanding was through code and not through formal definitions as addressed in this class. Lastly during the second half of the semester, I learned a lot about understanding big-O complexity in this class. I enjoyed learning about this even if I believe that it will not be very beneficial for much of my career. Big O notation in my opinion is more of a theoretical computer science focus and is not the pertinent in todays world were changing code bases can be more resource intensive as just using a less complex algorithm.

Lastly, throughout this semester I have grown as a mathematician in that I have a deeper understanding of set notation, graph theory, and structured mathematical proofs. During these past couple months I have increse my knowledge of mathematical notation immensely. Although my professor, Brendan Mumey, did utilize set notation in many of his lectures I was quite lost but with this new gain knowledge of many specific notations I have been more comfortable reading mathematical equations and books. Along with this, I have increased my knowledge of graph theory during this semester. I have concrete definitions of many concepts such as trees and forest now in my mind and plan to use them in future courses. Additionally, I now can say with confidence I atleast know the basics of some mathematical proof methods. These methods include but are not limited to proof by example, proof by contradiction, and proof by induction. For each of these methods listed I have included an example of a proof that I have created during this class below.

In conclusion, througout this semester I have grown as a student, computer scientist, and mathematician by increasing my fundamentatl knowledge of how mathematics are tied directly to life. Along with this I look forward to increasing my knowledge of theoretical computer science in this upcoming semester and applying what I learned in class to problems outside of the realm of education.

A Proof by contradiction

Use a proof by contradiction to prove that if an edge is removed from a tree, then the resulting graph has two connected components.

By definition a tree must be acyclic and the number edges can be found by v-1 where v is the number of vertices.

Our statement above is false at n edges and n+1 vertices, minimum such. But at n-1 edges, n vertices our statement is true in that if the edge is removed two components will be found in the resulting graph. This resulting graph after removing can be expressed as having a total of v-2 edges and still v vertices. By previously stated definition, in order for a tree to remain a tree after manipulation it must be acyclic and the number of edges must be v-1. Therefore our previous tree no longer fits the definition of a tree unless the original tree is split into two connected components. Along with this when splitting a tree at say previously connected vertices v_1 and v_2 since by definition there is no cycles, these two vertices are no longer connected in any way by an edge or it would have not been a tree to begin with.

Therefore we can conclude that if there is a tree G that has an edge taken away the resulting graph will be two connected components.

A tree is an acyclic graph which means it has no cycles and it is connected. Following this definition a tree has to be a simple graph.

We know a graph is a tree if there is absoluty 1 path between all pairs of the vertices

Let n = a graph such that there is exactly 1 path between all pairs.

Therefore n is connected, meaning no cycles

Let a and b be verticies of n

Between a and b there are 2 paths which is the contridiction

The path between this pair contains a cycle which contradicts one of our proproties of a tree (A tree is an acyclic graph).

B Proof by Induction

Prove that the loop invariant is when entering the i^{th} iteration of the loop is, "the variable CURGUESS stores the second largest value of $A[1, 2, \ldots, i]$, and CURMAX stores the largest value of $A[1, 2, \ldots, i]$."

CURMAX is the maximum and CURGUESS is the second maximum in $A[1, 2, \ldots, i]$ being I[i] and this is the induction hypothesis.

$A[i + 1] = CURMAX$

$A[i + 1]$ is greater than preceding elements and as such $CURMAX = A[i + 1]$ and among original elements $CURGUESS =$ previous max.

Thus $CURGUESS = CURMAX$ and $CURMAX = A[i + 1]$ which satisfy the condition.

$A[i + 1] > CURGUESS$ but $A[i + 1] < CURMAX$. Therefore the largest element is still $CURMAX$. But not the update will change the $CURGUESS$ to the new element of $A[i + 1]$.

With this we can now state, $A[i + 1] < CURGUESS$ and that for either $A[1, 2, \ldots, i]$ or $A[1, 2, \ldots, i + 1]$.

Thus, $I(i + 1)$ is true and the postcondition that $CURGUESS$ is the second largest element in Array A is fulfilled.