CSE 373 – Blog Post

Ainsley Lai

March 2022

1 How has your understanding of CS changed?

1.1 Project of Review

Huffman coding was the final project of my CS 211 class (equivalent to CSE 143 at UW) when I was at Bellevue College. In a nutshell, Huffman coding is basically a way to compress/decompress a file by assigning smaller values/bytes to more common characters so that the file takes up less space.

While I will admit that I don't remember the specifics, the main way of implementation uses binary tree to build a binary string of 1s and 0s where the left branch adds a 0 and the right branch adds a 1.

1.2 Design & Implementation

Looking back on my education in that course, I was only exposed to the idea of the binary search tree and thus didn't really consider how mutable the idea of a tree is. For instance, I had no clue about other types of trees like LLRB or ternary search trees. There was also the concept of choosing a data type that fits the functionality that we want and then using an appropriate data structure.

Though for this implementation, I think the tree is actually fine since the implementation already has a map inside it to store character-frequency pairs for compression. The tree it uses though, reminds me of the min-heap since the levels of the tree are decided by priority whereas Huffman is kind of the opposite; the overall root node would have the highest 'priority' value. Going with that line of thinking and exploring other possible data structures, I suppose it might be possible to use a hash table and function that relies on character frequency instead? I think generally, hash tables/maps are faster than trees so it may be better actually. There are likely a lot more ways to implement Huffman coding but I'm just one small brain that can only think of so many ways.

1.3 Analyze & Compare

In the course, Big O notation was the premier way of analyzing runtimes along with old-fashioned experimental analysis. Before 373, I had no clue that there were other things like

Big theta and omega asymptotic analysis. From a teaching perspective, if I had to prioritize teaching one way of asymptotic analysis, Big O would probably be my choice as it'd get at the most important case: the worst case. Asymptotic runtime analysis does seem to get at a larger range of descriptions from best, average, to worst.

Reflecting about coding as a whole, it might seem to many that the most important thing is for the code to accomplish its task. While this is true, completion of the task is not the only thing that matters in the same manner that ends don't always justify the means. The way that code is written and how it goes about performing tasks matters as well, especially in the context of runtimes and optimization. Some algorithms go about performing tasks faster than others. On top of just considering how efficient algorithms perform, we should also consider how it goes about performing its tasks. The next section goes a bit more into this.

1.4 Critique & Connect

To many, it's obvious that code are the commands given to computers and helps drive the actions of technology but having grown in my technical maturity, I've been exposed to more examples of how much weight these commands have on our lives. Perhaps the most apparent example lies with topics in computer vision and graphics. For example, there've been many articles writing about how autonomous cars (and lots of tech that detects people) have harder times detecting those with darker skin. In the context of cars and traffic, there is an obvious problem at hand that we need to find a solution for. In order to be safer, detection of people has to be accurate regardless of skin color. We cannot force them to bleach their skin just so that cars won't hit them. This example appears in our seam carving project as well. In dark pictures, those with dark skin would be more prone to being "carved outthan those with light skin. However, this is a really hard issue to solve if we are only deciding things by color. Algorithms might have to be changed to use some other parameter than color for the energy function. Sadly, this might just be a fact of life and a negative part about the technology we use, just like how stairs aren't friendly to those whose legs are impaired.

On a more biased note, the priority queue project presents another point of critique. It seems natural to be able to order and flag content for moderation but the underlying algorithm that decides the priority is prone to human bias and what different people may consider 'misinformation' or offensive content. By implementing something like that, we cater the system to work for one party and not the other and it begs the question: should we be doing that?

The DuckDuckGo browser's CEO recently announced that it is down-ranking sites associated with Russian disinformation". Observing the Twitter thread, a lot of people are mad at this decision as the main crowd that uses DDG are those who want a private, unfiltered, and uncensored internet. Especially in a time of war with propaganda on both sides, we must ask ourselves, how and who decides what is misinformation?

2 What part of the course most changed you?

The part of this course that was most meaningful and interesting to me on a personal level was learning about affordance analysis and design justice. Through my past CS education, I have not had any professors tell me about such things. It was interesting hearing about Kevin's take on ethics and taking it into my life and growing my own views on ethics in technology. With these ideas in mind, I believe that I will view data structures and algorithms with a more critical eye. Focusing on not only the technical aspects of my code but also keeping in mind the greater social implications and impacts it can have on its stakeholders. I believe that this course has made me keener on thinking and analyzing the inherent flaws and injustices that can exist in data structures and algorithms.

3 What's next?

Spring break and me doing literally nothing! Then, spring quarter taking AMATH 353, INFO 350, and JAPAN 103. I also got a position as a student leader at the UW Bothell GenCyber camp in the summer but I'm not sure if I actually got the position since the guy in charge of it didn't respond to me when I replied to his email saying that I accept the position. I also interviewed with Seattle City Lights for an Information Managment Internship and they complimented my resume design (which was a template) but I'm not sure if I really want the position since it's for a whole year and needs 20/hrs a week during the academic year and 40/hr a week in summer. It would be good experience so maybe I should just suck it up and accept if I get it. Also, Elden Ring looks like a cool game so I might play it and ask to borrow my brother's computer to play it.