Alex Childers

11/04/2018

PA #4 Reflection Essay

If I were to rate my work on this assignment, I’d give it a 5.5 - 6.5? out of 10. I’m pretty happy that I was able to overcome some of the challenges described below, but I also wish I had done the extra credit work to decompress a file. I think it would have helped me learn more, and I might do it later anyway, just for fun. I also think I should have spent more time implementing functions in my WordCounter class, rather than having all the encoding performed in main.cpp. It would be nice to have a header file that contained both encoding and decoding instructions, so that main.cpp was simpler and easier to follow.

I actually found this assignment easier than the last couple of assignments. As you can see in the header of my solution, it only took me about 8 hours to complete this assignment, compared to the 10+ hours I’ve spent on other assignments. Part of it was that, from studying the labs, I could picture how to use an unordered\_map in my head, as well as knowing that I could generate unique binary numbers from the functions we wrote in Lab 9. Additionally, I felt that, thanks to the algorithm being provided in the assignment description, I pretty much knew what I needed to do for the assignment – the time I spent on the assignment was just figuring out *how* to do it. I spent a couple of hours at the end debugging a pretty small issue as well, so in reality, the code would have been functional enough to turn in after about six hours of work.

There were two main obstacles that I overcame when working on this assignment: generating unique binary numbers, and tracking new line characters in the input file so that I could write them to the output <source>.compressed.txt file.

I spent some time trying to picture how I was going to generate unique binary numbers for each word in the file. I noticed that the numbers in binary incremented for each word that was removed from the max-heap. With that in mind, I used some scratch paper to work out how I was going to generate binary numbers in this manner, thinking about it from a “how can I convert one string to another string” standpoint – e.g: How does the string “11,” or 3 in binary, become the string “100,” or 4 in binary? After some time, I decided to revisit the functions we wrote in Lab 9, which generated binary strings, to see if those functions could help me. I realized that Task 4’s function essentially converted an integer from a base 10 number to a base 2 representation of that integer in the form of a string. I hesitated to use it due to algorithmic efficiency concerns, but I decided to use it anyway because I couldn’t think of a more efficient way to generate unique binary numbers.

I also spent a couple of hours trying to figure out how to track new line characters. After I had finished the bulk of the assignment, my program was coding and compressing the input file correctly, but it didn’t insert new lines into the .compressed.txt file, as your example files did. I tried to insert new lines by prevented new line characters from being put into my frequency hash table, but this caused some extra problems because spaces and empty characters were being erroneously counted in the .codes.txt file. I was eventually able to solve this problem by directly editing the StringSplitter.h file to prevent empty strings from entering the vector that its split function generated. Then, I used an if check every time the program operated on the frequency hash table – if the next word in that vector of split strings was a new line character, I would skip processing it so that it didn’t enter the frequency hash table, and when I was writing to .compressed.txt, I would insert a new line character into the output file stream.

My favorite thing about this assignment was that, since it’s apparently comparable to the process used in the ZIP algorithm, I felt like I was programming something significant with this assignment. Similar to the last assignment, where I could say “I made a game that learns more animals the more you play it,” I can say for this assignment that “I wrote a program that compresses text files,” and I think that’s a really cool thing to do!