Samuel Bailey

CS-260

Final Project Reflection

1. Data Structures
   1. **VECTORS**: In week 4 we are using vectors. Vectors are great and easy to use. They allow user defined data types which makes vectors very programmable. Vector require minimum code to setup and it easy to implement search algorithms. Like the ones made in week 4.
   2. **HASH TABLES**: Next we have hash tables. Hash tables have keys representing each hash. Which makes sorting instant or O(1) which is constant being the best time you can have. We can see how this is implemented in from week 5. A downfall of hash tables are that you have to implement a logic to free up space once a class is destroyed. At it’s root a hash tables is array’s, that’s why it needs to have the space freed up after deleting.
   3. **TREE STRUCTURES**: Tree structures are a little tricky at first but once understood relatively simple. Tree structures are the only data structure that is not linear. This structure has a root value as well as a subtrees called children in most cases. In week 6 we used this data structure and as long as it is implemented correctly(which it is) then it is one of the best data structures. The downfall of using this is if the data becomes mismanaged the tree is hard to use.
2. Algorithms
   1. SEARCH: Each week we use an algorithm to search the data given to reach a result. The search methods change depending on how we created the data structure. When creating these searches, we always want to keep in mind what will be the most efficient as well as the best amount of space/memory. All the search algorithms give us exactly what we need. Usually, search is done with either a key value or searching through the title until a march is found.
   2. SORT: If we take a look into week 4 we can see multiple sorting algorithms. Starting with quicksort we have which uses recursion to sort the data. Next, we have selection sort. Selection sort works by defining a point then moving a pointer through the data and switching either to the left or right side depending if it is less or greater.
   3. HASH/CHAINING: In week 5 we can see an example of using hash tables to represent and store data. Hash tables have a set key with every section of data, so it makes it very easy to search through. In this project we also have chaining implemented which is a LinkedList of an array of data. This is particularly when hash tables become confusing. It’s best practice to keep each data set separate that way using search and sort works more fluently.
3. Student’s Choice
   1. Each program was made according to the grading rubric and meet all the required steps. Each one was made to the best of my ability and knowledge of the subject.
   2. //
      1. Each project had different sections, where the code is split up. Creating code this way makes debugging much easier as well as simple readability and understanding. If we look at week 5 we can see how everything is split up by comments explaining what is going on.
      2. The code created is also reusable, we know this to be true by reusing it the weekly projects. We simple kept building from week 3. If we wanted to reuse the code in a different project we could use some of the methods and functions if we include the project file in a different file.
      3. Writing annotations/comments are one of the most important parts of programming. The comments written in the projects explain exactly how/why something is the way it is.
4. Conclusions
   1. Data structures are the MOST important part of programming. This is what sets apart people that know how to program vs someone that has watched some YouTube tutorials/taken an entry level class. We see throughout this term each week a new data structure and the efficiency and effect each structure has.
   2. To be clear, an algorithm is a set of instructions. When we give a computer a set of instructions we want the computer to respond quickly and promptly. As FAST and takes the LEAST amount of memory as possible. Our choices is choosing algorithms directly effects both of those things.
   3. Data structures are fun, and dictate entire projects. Something that I worked on this term was a contact application. The application allows users to input data(name, #, and birthday) the app was built with a struct of vectors, this was about as simple as I could make it. I saved the data to a txt file and loaded the data upon starting the program. The program was working but had a fair amount of bugs and just needed a lot of work. Ultimately I didn’t have enough free time to continue to work on it and will pick it up in the future.