Fellipe F Lopes

CS 163 – Writeup Assignment 5

The data structure assigned was a graph. The graph performed nicely, allowing sorting the information effectively and retrieving items quickly comparing the data being searched to the data in the nodes; using the divide and conquer technique. The fact that each node has to pointers, to an index in the graph and a character pointer, makes the insertion much easier than before.

At first, reading the assignment requirements I had the feeling that a tree would in fact be the best resolution for this data and how it needs to be handled. The binary search allows quick access to a place with the same index, and afterwards, access to the information. The information stored in a structure makes it easier to be listed. The main data is: item name, color, size, description and website where such item is stored.

The efficient part about my program was getting all the information from main and putting it into a structure that was later on passed into other functions to be manipulated. The functions that do not have arguments are public members of the class, being able to receive information from main and able to use information in the private section. The add edge function is responsible for adding a “choice” to how the person is going to access the place they are going to. The edges point to the vertices when they are connected. The array for the edges and the pointer to the vertices exist in the nodes of the linear linked list. The add edge function will determine if there are nodes in the linear linked list at that index, if not a new node will be created and the temporary node passed as the head for that linked list; if there are other nodes in the list, the temporary node will be added at the end, pointed by the next pointer of the last node.

The most difficult and challenging part in this program was making the destructor and remove all functions. I feel like these functions require a lot of moving small pieces, and principally deleting the item associated to head next, pointing it elsewhere for the same process to happen again. As for the remove all function, the process was a little more complicated to figure out. Using recursion to execute this process made me stop and think ahead all the work the function will do before the deletion itself happens. In addition to that, the add function gave me some issues during execution. The first time I ran the program it was giving me a weird segmentation fault not allowing new nodes to be inserted. After investigating the issue, it started working normally, and all the nodes added as normal.

If I had had more time to finish this program, I would definitely have implemented the export to file and import from file functions with the concepts learned last term in CS162. It would make it possible for the user to export all the information to a file, saving their imports. This function would also check the current directory for a text file, and once it was found, all the information would be brought over to the current program execution, calling the add function for each of the items found. Another function that I would have implemented as extra was find one. This function would be responsible for finding a certain edge and showing all the information contained in it. This function would be a Boolean type, informing main, about any errors that might have happened during execution.