

## Efficiency Write Up – Project 1

For my program I had Info as my superclass which contained similar information like name, location, and quantity. Then for my subclasses (relief types) I had food, clothing, and shelter. I had one more level of inheritance where they all inherited food node, clothing node, and shelter node. From there I had a list class which contained three node pointers pointing to the three separate circular linked lists. Finally, I had a history class that had a name and frequency integer, this was responsible for keeping the history of what the user searched for. If I did this project over again I would've changed the hierarchy, instead of my nodes inheriting the classes with data (food, clothing, and shelter) I would've made it a containing relationship. I believe it makes more sense for the node class to just hold the data instead of having the data. But having this inheritance taught me how to access the functions and data from within the hierarchy.

As far as my debugging experience, GDB was a huge help in determining issues I was having from the early stages to late stages. In the early stages I used GDB's lay source function to step through my code and make sure the right constructors/copy constructors were getting called. The biggest issues I had was when it came to adding a node to a list. I was using my copy constructor from my node class with the keyword new. Then in my add node function was using new again to create another object. I believe the one I had created in main was not getting passed properly to my add function. So the values showed up as garbage because it was going off the object I created in my add function. This was a difficult issue to solve because it wasn't inherently clear what was happening, even stepping through the code it seemed like everything was working.

Another issue I had was when I was destroying my lists, in the remove all nodes function I passed in the front of the list then disconnected my CLL then, using head recursion, deleted on my way back setting each node to null. The problem was after I had done that it had memory leaks. But using gdb I was able to determine that this->rear was not getting set to null which was the culprit.