Emmanuel Rodriguez
EEL4851C.001F08
Assignment 3

Purpose and Objective:

The purpose of this project is to learn how to implement and use the Heap Tree ADTs. Also, learn about depth-first and breadth-first searches, and also using make files. The make file is important for source code management, since we will have many source code files. Also, we will learn about connected component based coloring operation in image processing. Documenting programming experience and describing the program will teach how to effectively communicate the program's intentions with either a programming professional or any other person.

Code Compilation/Run:

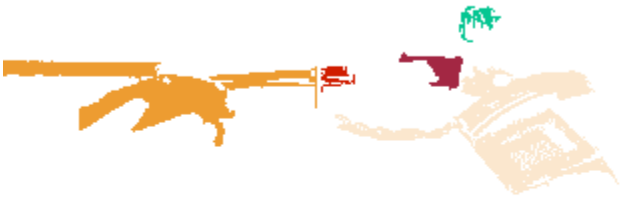
The first step to running and using the program is compiling. For this assignment we wrote a make file to compile all of the source files for this project. To compile the project type 'make'. This will compile all of the .c source files at the same time along with the necessary libraries. The make file will call the executable 'application'.

The final step is running the program. To do so, you type the name of the executable file, whose default name is a.out. However, since the –o operand was used to rename it, 'application' should be typed. Also, this program takes six user inputs through the command line, which should be typed on the same line as the executable file name. Those inputs in the required order are the sample color file, input image file, output image file, threshold value, component size, and search type value. "application YellowHandSample.ppm MarcelDataSet_Push_Frame12.ppm ThresholdOutput.pgm 6 20 1" is an example program run.

Results:

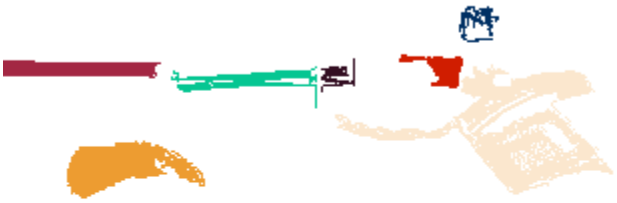
MarcelDataSet_Push_Frame30.ppm – Threshold 5 – Component Size 70 – Yellow Sample

MarcelDataSet_Push_Frame30.ppm – Threshold 6 – Component Size 50 – Blue Sample



MarcelDataSet_Push_Frame1.ppm – Threshold 5 – Component Size 40 – Yellow Sample

MarcelDataSet_Push_Frame1.ppm – Threshold 6 – Component Size 100, Blue Sample



Bugs/Errors:

The first time I wrote the code it was not modular because I was directly accessing structures when I could have used function already written to do the exact same thing. My finished code is modular and as a result is smaller.

I also encountered a segmentation fault because I used "=" instead of "==" in my HeapEmpty function.

```
 Saving component of size 2930
Program received signal SIGSEGV, Segmentation fault.
0x080494cc in ListLength (L=0xbfffeec0) at GeneralizedListImplementation.c:24
24          Current = Current->next;
(gdb)
```

Conclusion:

I encountered many more error throughout the process of writing the program than stated in the previous section. But, they were minor in comparison to those which I have stated. Syntax errors resulting from a forgotten semicolon or misspelled function names were prevalent. The only thing new I learned from this project was the heap tree ADT implementation. However, the project has also helped gain confidence is using functions as parameters.