Programming Assignment 4 – Approach

Luke Orth

CSIS 215-B01

My approach to this assignment is to create separate header files for each self-organizing heuristic. I’ll then inherit the SelfOrderedListADT for each heuristic and create its corresponding implementation in each of these header files. I expect that each of these will be relatively similar. Getting the size of the list with (**size** function), printing the list (**printlist** function), and getting the number of compares (**getCompares** function) should all remain identical in each implementation. Even the **find**, **add**, and **reorder** functions I expect to mostly recycle for each heuristic.

I’ll begin by developing the **add** function. This is because it’s needed for every other function (you can’t do much with an empty list). The next logical progression seems to be finding a specific value that’s been added, so building out the **find** function will be my second objective. After I’ve constructed the ability to add to a list and find values, I’ll incorporate the **getCompares**, **size**, and **printlist** functions. These should all be straightforward to implement and will be good for troubleshooting purposes. Finally, I’ll tackle the **reorder** function. I expect this function to be the most complicated to build.

I’ll build and use my **Driver.cpp** file to test these functions as I go. Once everything seems to be working as expected with manual tests, I’ll run the “test.txt” file through it. If the results are satisfactory, I’ll finish it off by improving the UI a bit and adding extra comments where appropriate.

*Please Note ~*

*The executable file for this program can be found in the zip file at:*

***Luke\_Orth\_Self\_Organizing\_Lists\_Assignment / Debug / Luke\_Orth\_Self\_Organizing\_Lists\_Assignment.exe***