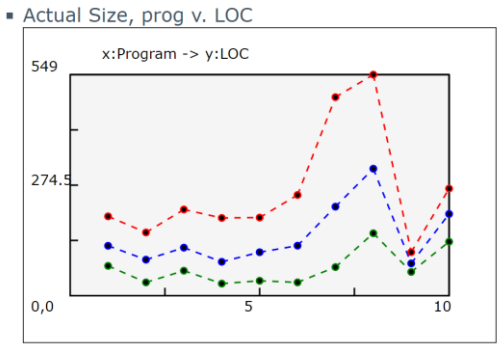
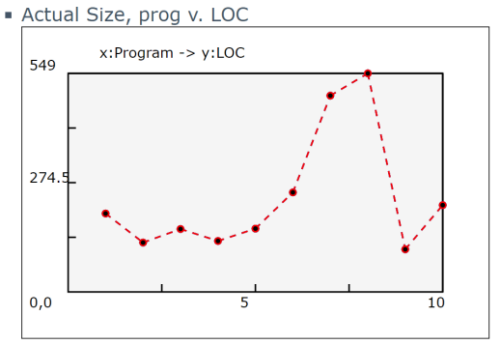
Analysis of Size Estimates

Software Size Estimates is try to estimate the size of a software application to make it able to manage the whole software project [1].
There are a lot of software size estimates method. One of the most famous method is Lines of Code (LOC). LOC also known as Source Lines of Code (SLOC) is a method to estimate size by counting the totals lines of the source code. LOC also used to measure the effort of the program.

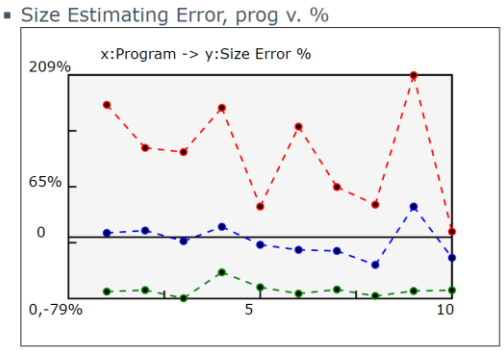
In this semester's SPI course, I have built 10 program, the size of each program of the wholes class and myself is show below.

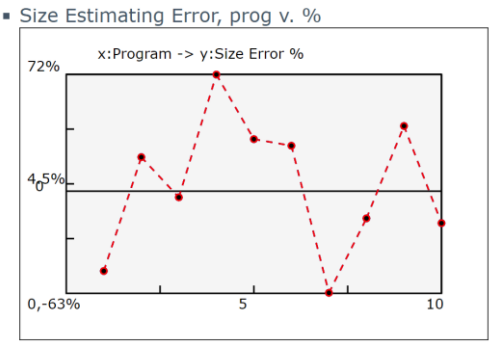




Class Myself

The size estimate error percentage of the class and myself is shown below:





Class Myself

As shown in the diagram, the size of my code most reach the top of the class. But the size estimated error is mainly in 70% statistical prediction intervals except program04. The accuracy of size estimation is travel around 0 and it went closer to 0 later. As shown in the diagram, program04 reach 72% and program08 reach -63% but when it comes to program 09 and 10, the error is much smaller.

The code before program06 (not include) is around the class average size. But after program06 (include) my LOC is increasing up so quickly and finally reach the highest of the class. After looking at the source code of program01 to program05 and program06 to 08 I found out that the main reason cause the increasing is that I use the

whole source code before as import. I write several new function to make the old program function can be used in the new program.

The reason I didn’t just copy and paste the original code is that this will make the main function too hard to read. Keep the program in a good structure is good for code review and testing. I also write a test function in the tokens.java it may also add some size.