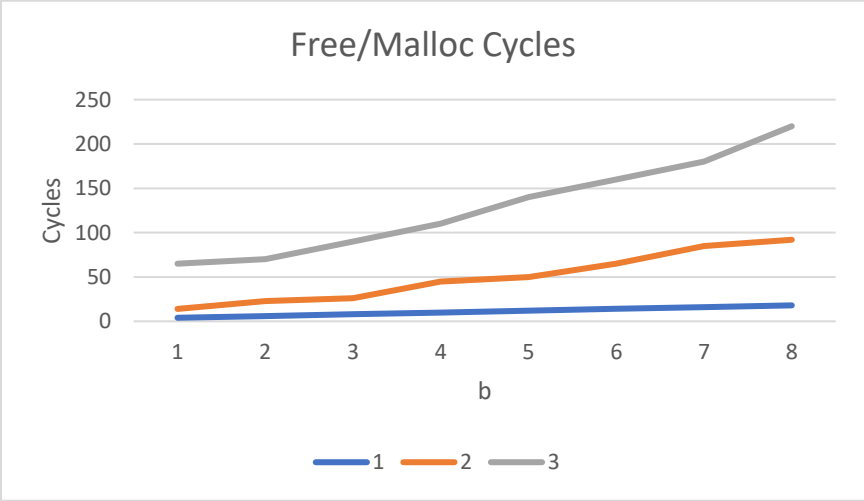
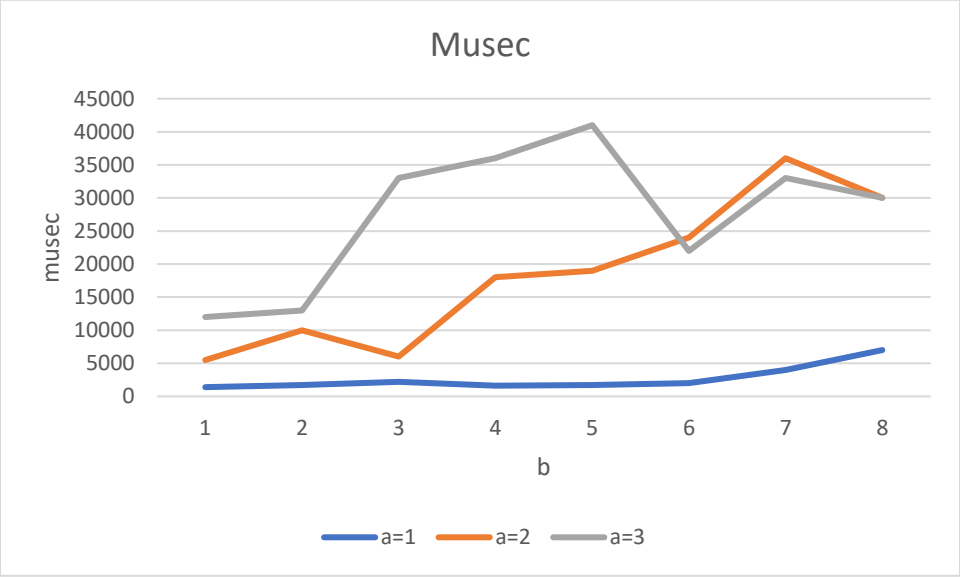
Austin Peterson
926006358

Buddy Allocator Analysis

The buddy allocator is designed to allocate and free blocks of memory while splitting blocks to create suitable segments of a smaller size and coalescing them back into bigger blocks for later use. The program takes 2 inputs, an A input and a B input, which affects the number of allocation/free cycles as well as the performance of the operations. While the data recorded for the number of cycles and runtimes vary to a great degree, average values can be observed and plotted to identify trends in the performance. As you can see below, increasing the values of A and B increase the number of allocation cycles across the board, with each different value of A representing its own line.



The values recorded for the program's performance varied to an even greater degree, however with average values the trend is about the same: as the values of A and B increase, so did the runtime (musec).



Austin Peterson
926006358

Unlike with the allocation cycles, the runtime dipped when A was equal to 3 and B became greater than 5. This could potentially be caused by the fact that program runs with a lower number of cycles have to perform more split functions to populate the lists of segment headers with properly sized blocks of memory. If there were an implementation that could initialize the list with a certain number of memory blocks to be allocated, fewer spits would have to be performed and the program could more efficiently return memory to the user.