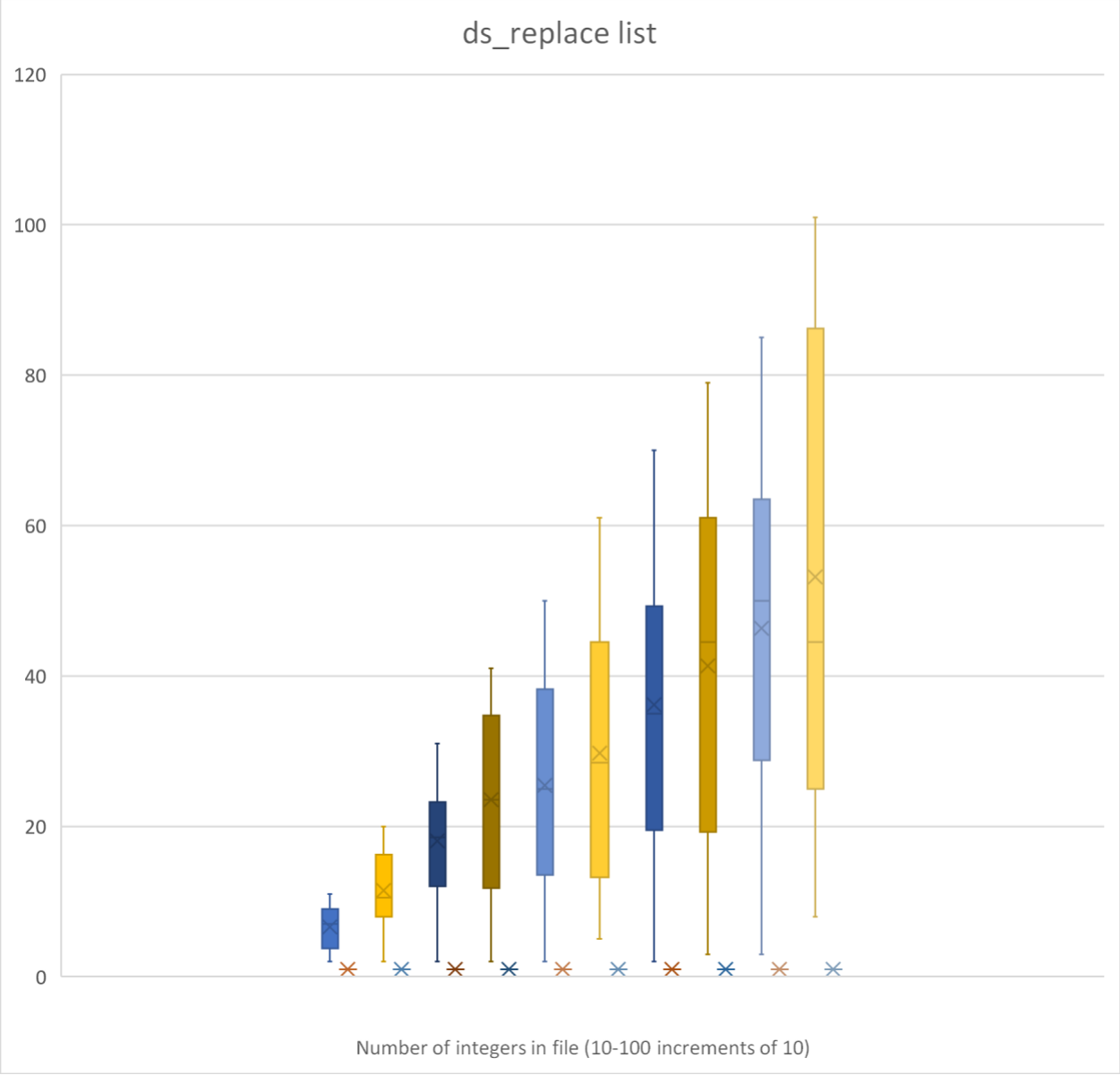
CIS2520 F19 Assignment 1
Stefan C. Kremer

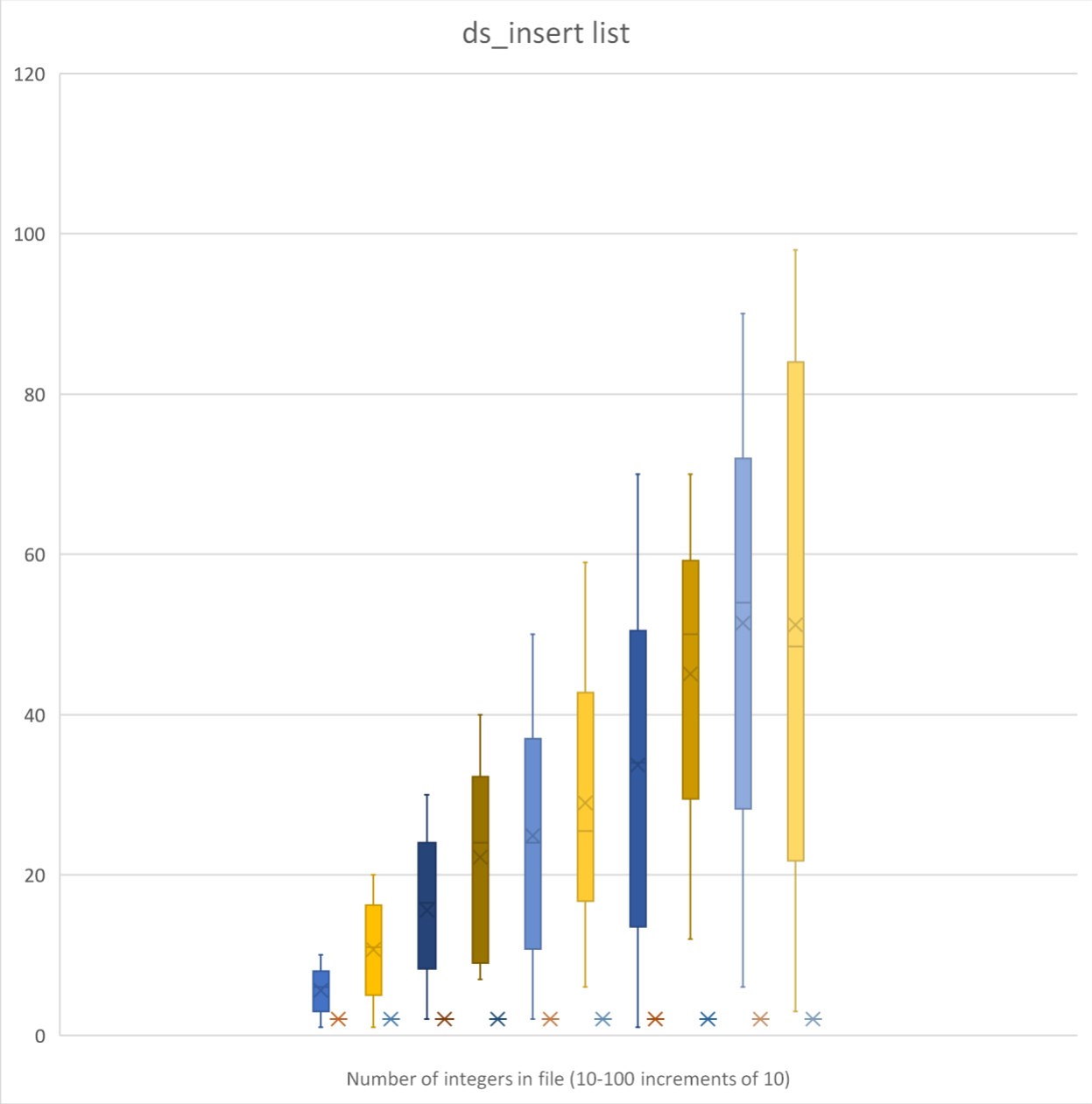
Aeman Abdulmuhssen
Student ID# 1003984
aabdulmu@uoguelph.ca

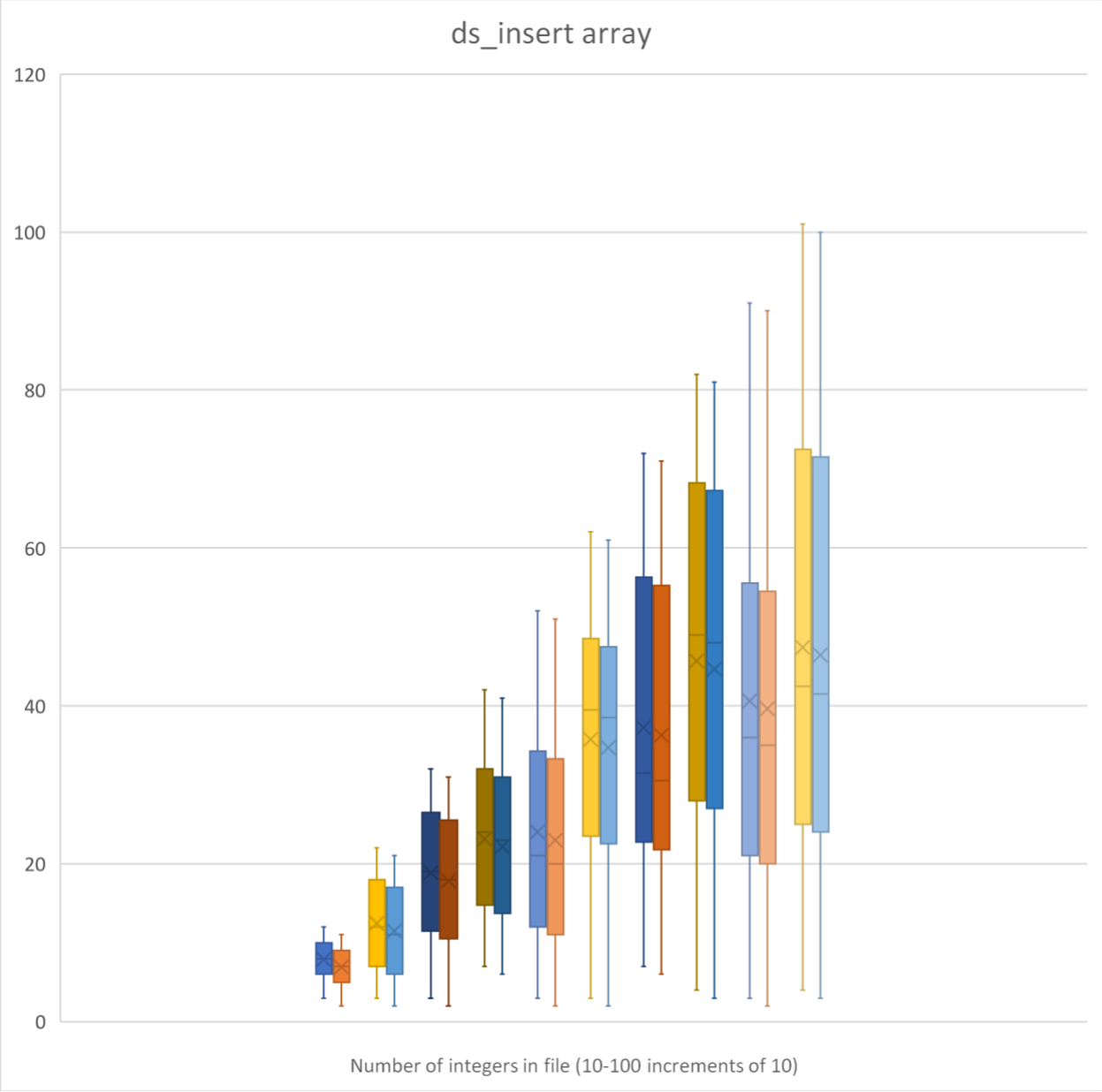
Part B – Analysis

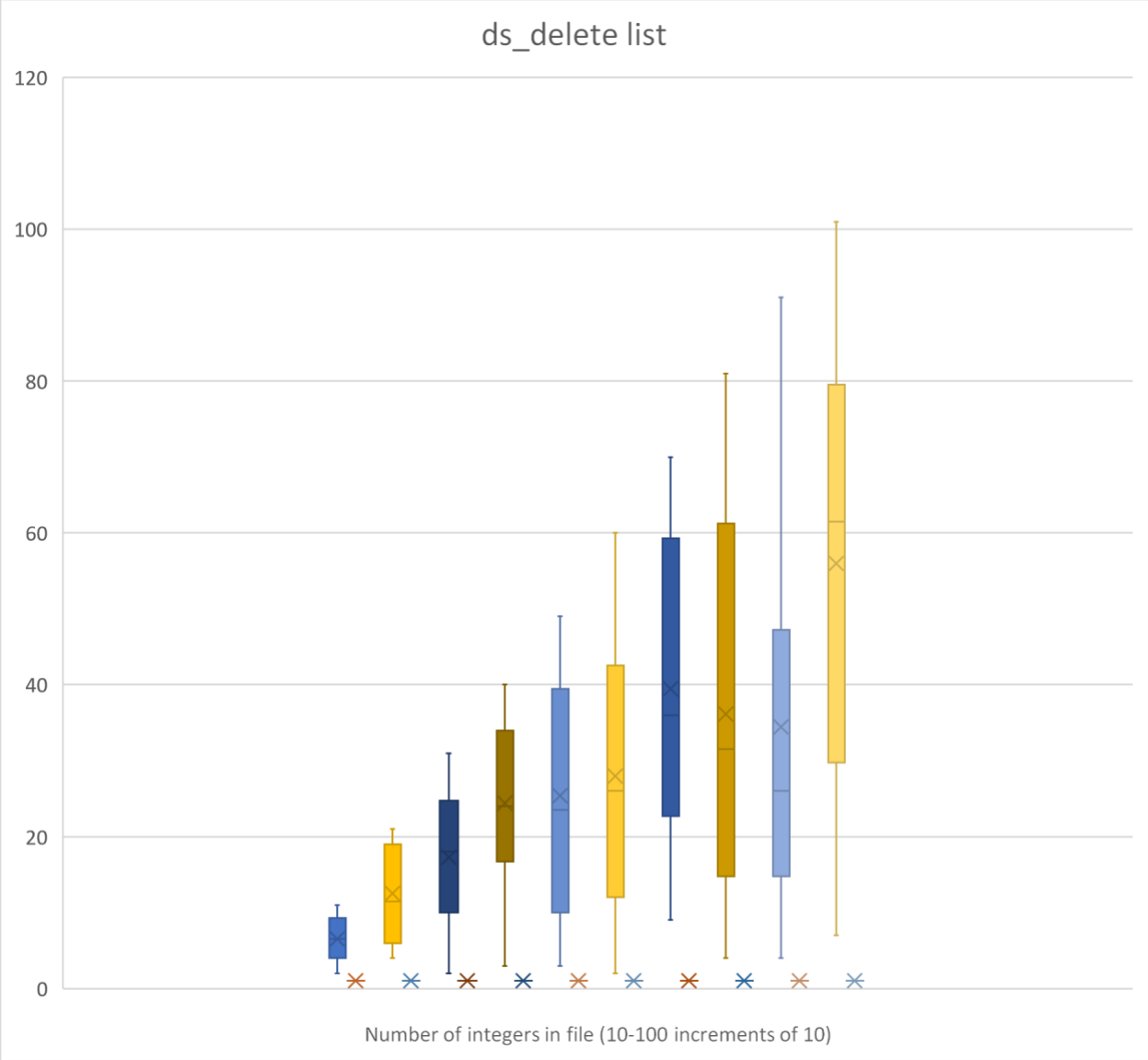
- For the graphs, they are arranged by number of integers in file from 10 to 100 by increments of 10. The boxes are arranged in Reads/Writes order. The max/min is shown by ends of each box and avg is shown by the line inside the boxes.

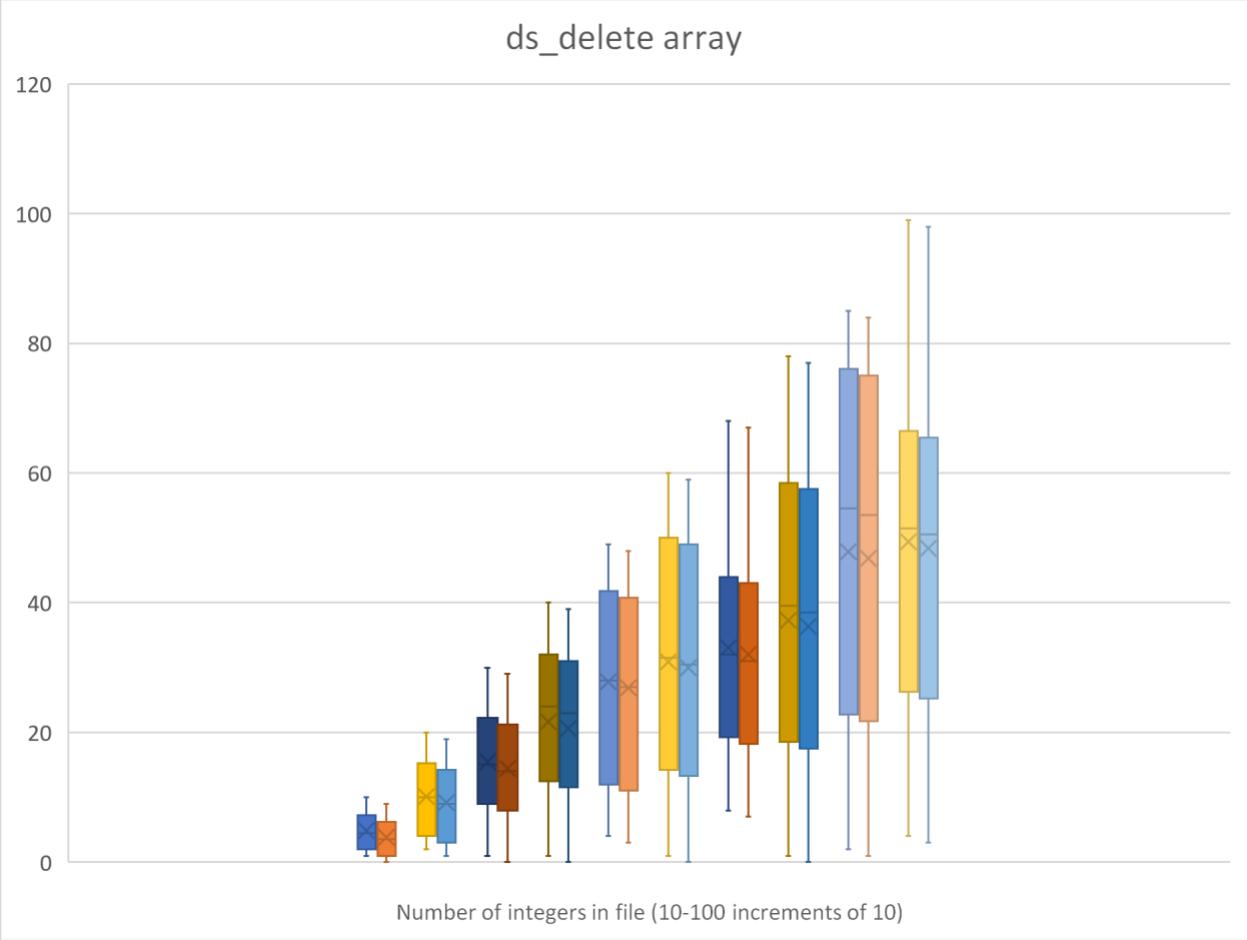


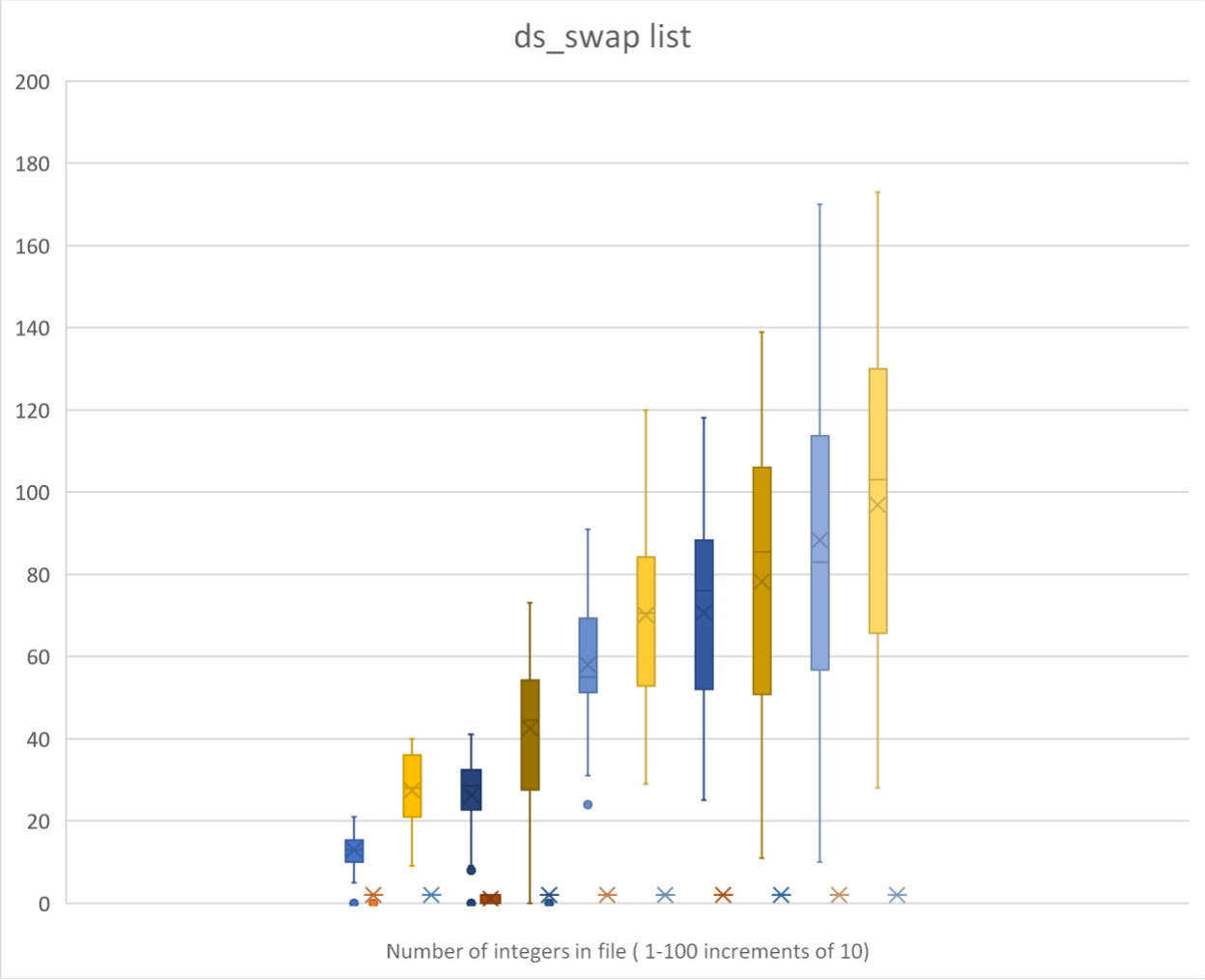

ds_replace array
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Number of integers in file (10-100 increments of 10)

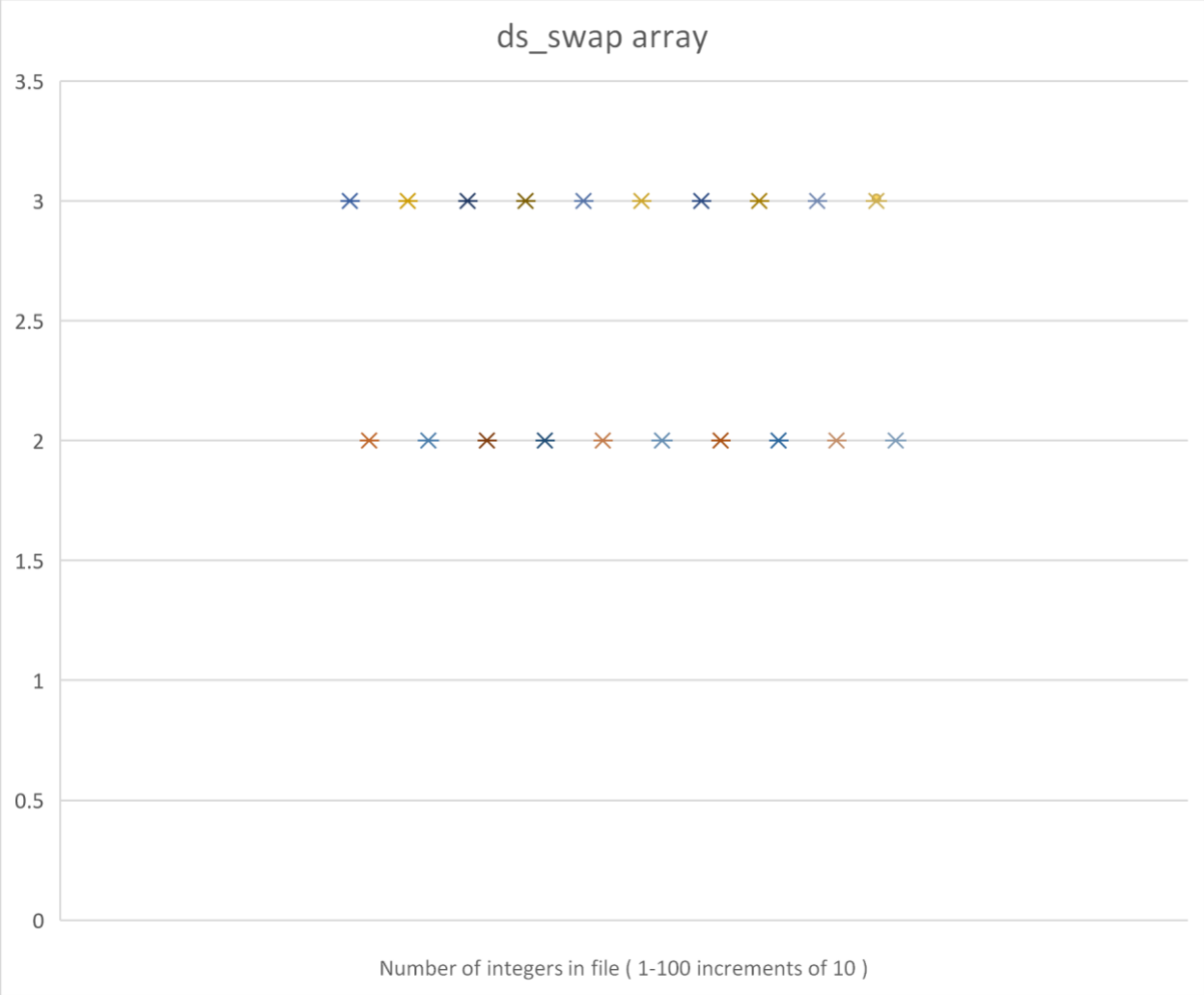

ds_insert list
120
100
80
60
40
20
0
Number of integers in file (10-100 increments of 10)


ds_insert array
120
100
80
60
40
20
0
Number of integers in file (10-100 increments of 10)


ds_delete list
120
100
80
60
40
20
0
Number of integers in file (10-100 increments of 10)


ds_delete array
120
100
80
60
40
20
0
Number of integers in file (10-100 increments of 10)


ds_swap list
200
180
160
140
120
100
80
60
40
20
0
Number of integers in file (1-100 increments of 10)



Discussion:

- Both data structures, linked list and array, are similar in how data is arranged but depending on the problem we are trying to solve one can be more superior than another. From the graphs above which are collected using our memory allocation program, we can indicate the complexity of each data structure when preforming functions like insert, delete, replace, and swap. When preforming delete and insert in a linked list data structure, we can indicate the number of writes in the program is 1. Unlike arrays, the memory after the index that is being deleted or inserted needs to be shifted making the algorithm more complicate and less efficient than a linked list. However, with swap and replace functions, arrays are more efficient due to direct memory access rather than sequential access in linked list. Swapping and replacing can

be accomplished with reading specified index and writing at that index, rather than traversing the list in order to find a specified index. To illustrate, linked list is more efficient and less expensive than an array to keep track of student list at a university/college or employees at corporate locations, as students and employees are easily removeable in case of graduation from the university or employees that no longer work for the company. This is the same case as in inserting new student and new employees if they are arranged by student/employee number. Arrays are good way to keep track of book record at a library. We can access specific book using an index and change the element structure from “rented” to “available” and vise versa.