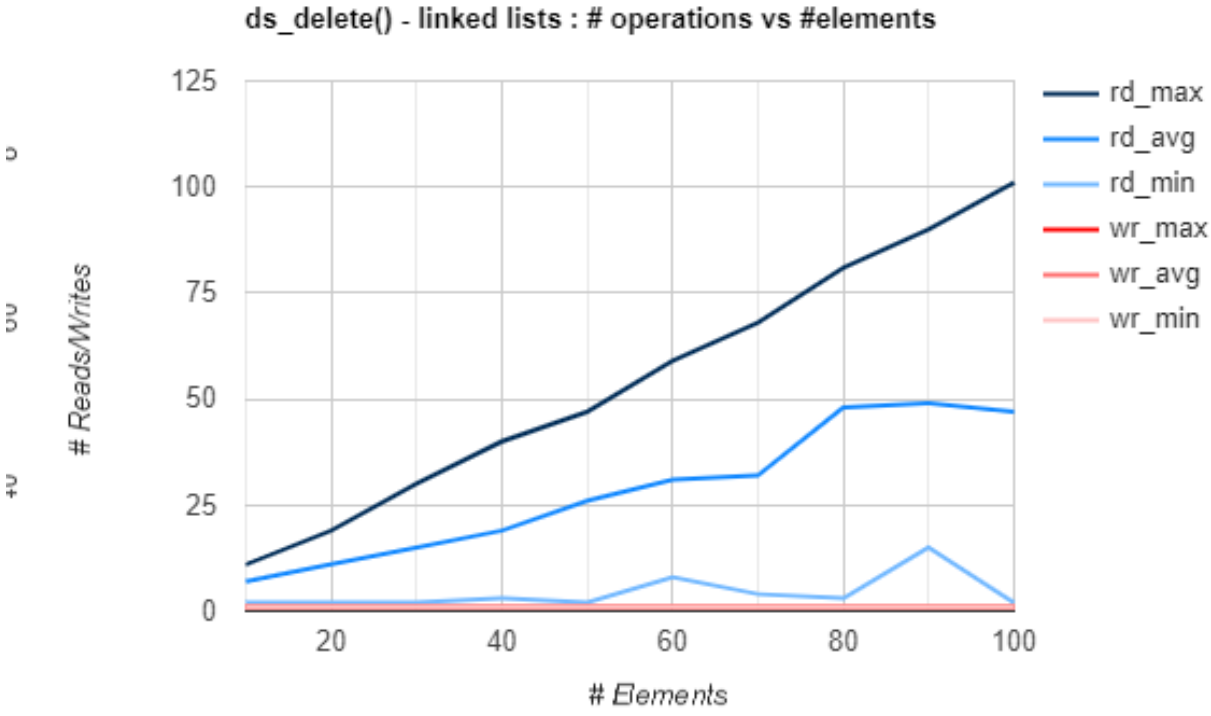
Lilian Shi

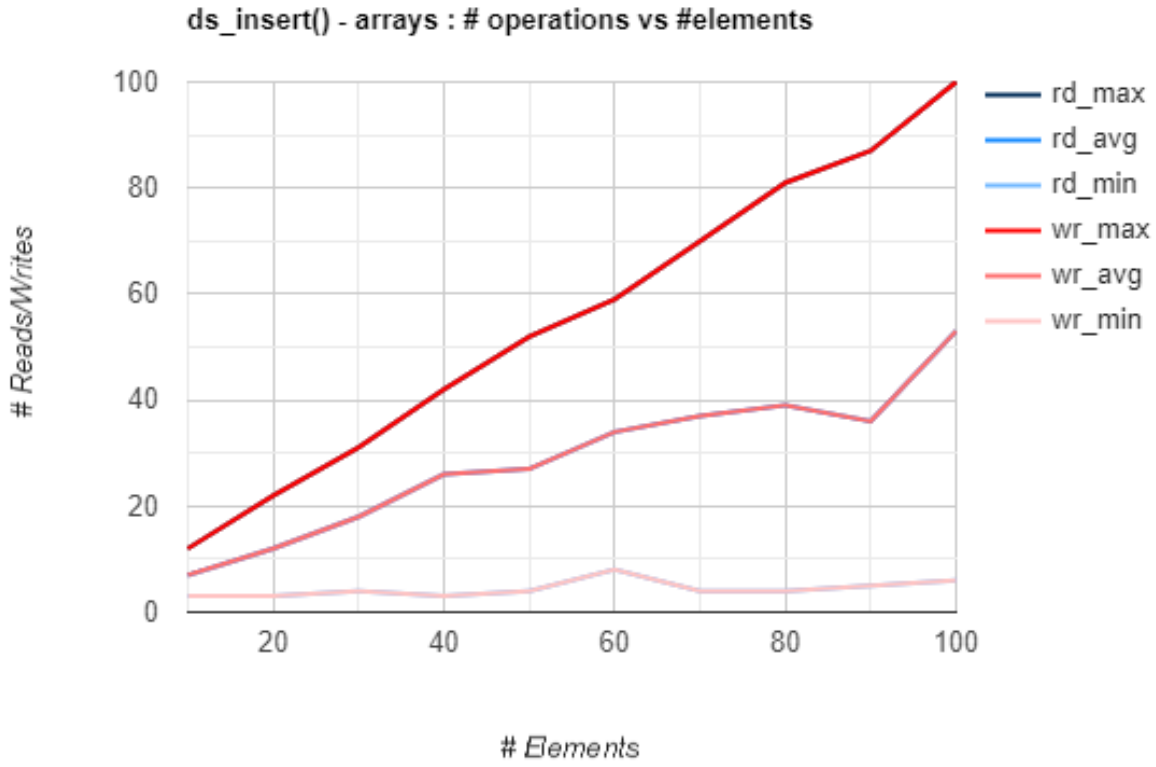
1048355

lshi02@uoguelph.ca

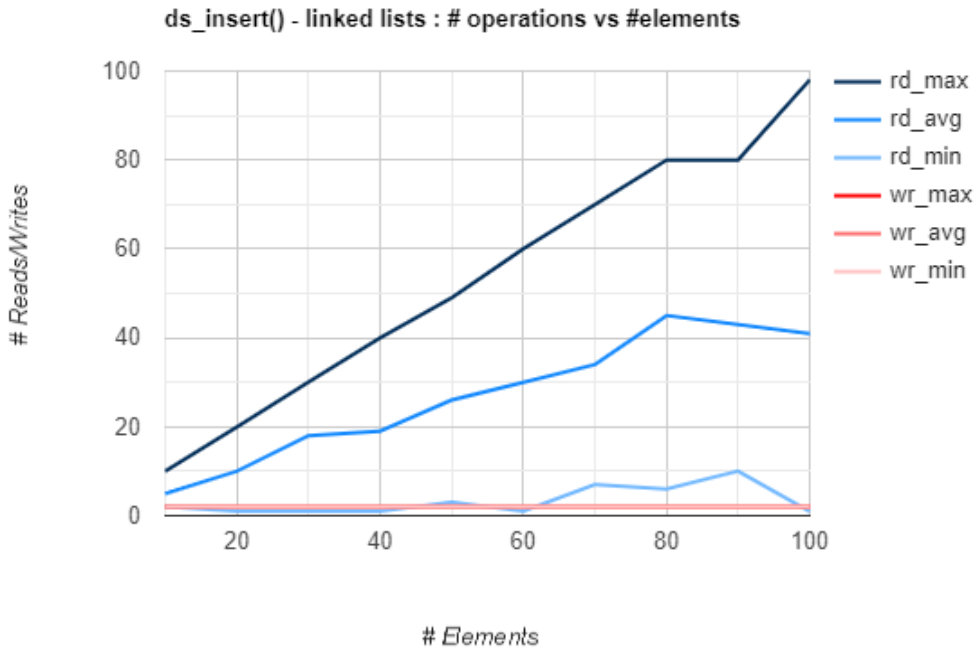
Above: max/avg/min were always equal for reads and writes which is why only three lines are visible



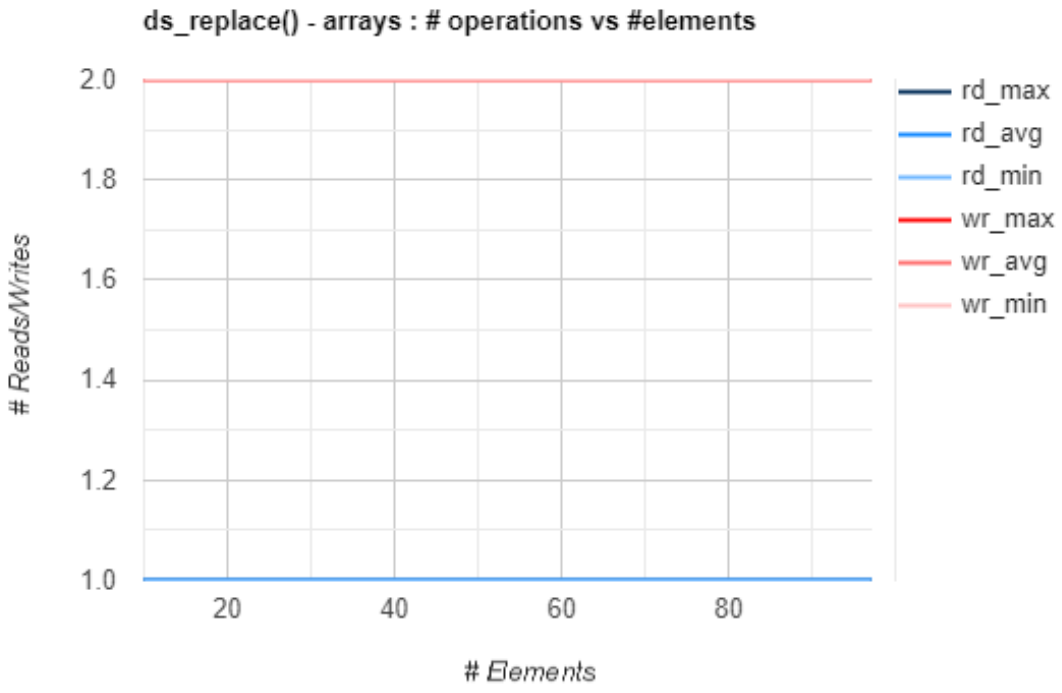
Note (for above): the max/avg/min read and write operations was always equal and constant



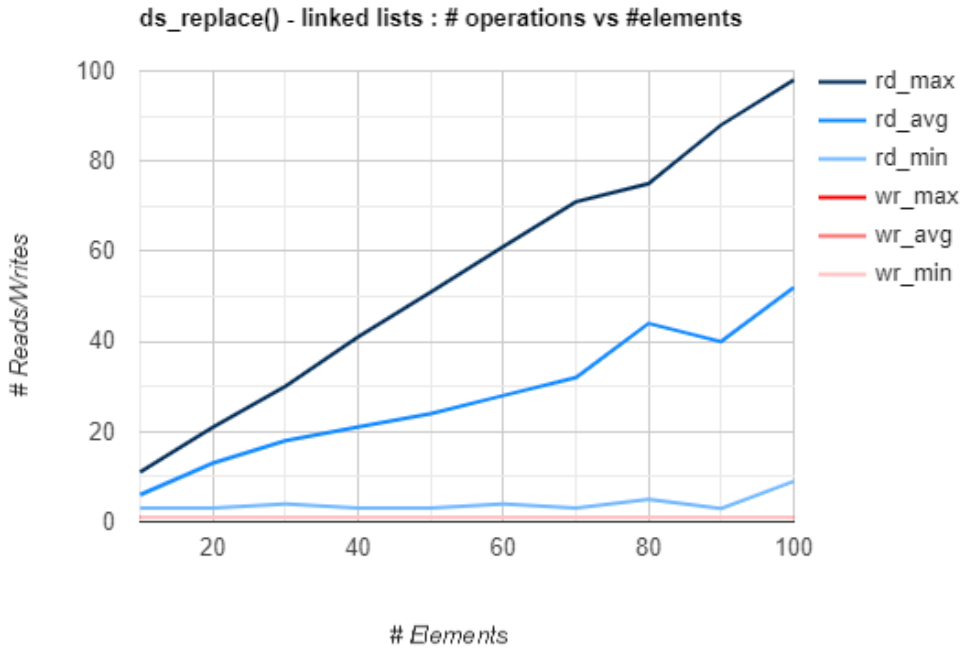
Note (for above) : the max/avg/min was equal for reads and writes.



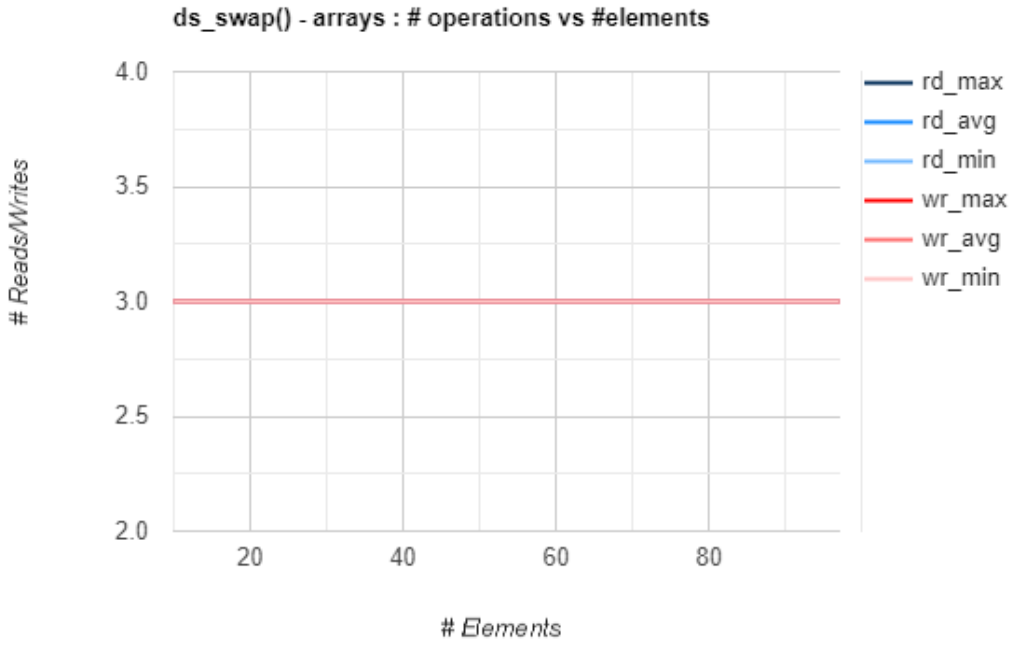
Note (for above): the number of reads was constant



Note (for above): the number of reads and writes was constant



Note (for above): the number of writes was constant



Note (above) : All values were constant

Above: all read values were constant

Discussion

Linked lists are more efficient in insertion and deletion operations, as they are not required to copy the remainder of its members after the element of interest (unlike arrays). As indicated by the graphs for ds_insert() and ds_delete(), the number of write operations for linked lists stayed more or less the same while theu grew ~linearly for arrays. Linked lists are probably more efficient data structures to keep track of things that may often change in the number of members they have:

e.g. Tracking hotel guests, members of a chat room, residents of a city

Arrays have more efficient performance than linked lists when replacing and swapping elements. Linked lists grow roughly linearly in read operations, while read and write operations are small and constant for arrays. Arrays would be a better choice than linked lists in cases where swap/replace operations are used frequently but the number of elements/members stay constant or change rarely:

e.g. Keeping a list of the top 50 songs, players on a sports team, someone's medical stats (i.e. every element would correspond with a different type of information, such as blood pressure)