Fiona O’Connell
Ayushi Sharma

Data Structures Lab 7

Running the file:

- The makefile should allow everything to be compiled and run with just

make
main

A description of the objectives/concepts explored in this assignment including why you think they are important to this course and a career in CS and/or Engineering.

In this lab we looked at three different methods to insert and remove items from an array. We then tested the efficiency of each method. In computer science, efficiency is important, especially when working with very large amounts of data. Understanding and being able to determine which algorithms work best and are most effective is very important for anyone looking to find a career in CS.

Descriptions and predictions for each of the three classes

The Array class is the base for the other two derived classes and holds the array of pointers used in all three. The Array class has insert and remove functions that add and remove linearly, starting from the lowest index and adding to the end, sorting by comparing and swapping with previous values to find appropriate place in the array. The remove function finds and removes the desired value, taking a non-pointer datatype as a parameter and comparing with array items, shifting to ensure there are no blank spaces.

The Center class has insert and remove functions that start in the center and determine direction to proceed after comparing with the center value.

The Blanks class leaves null pointers between values when inserting and only moves values when there are no blank spaces available at the desired spot. When removing, it does not make any move operations, only assigning the removed item’s value with nullptr.

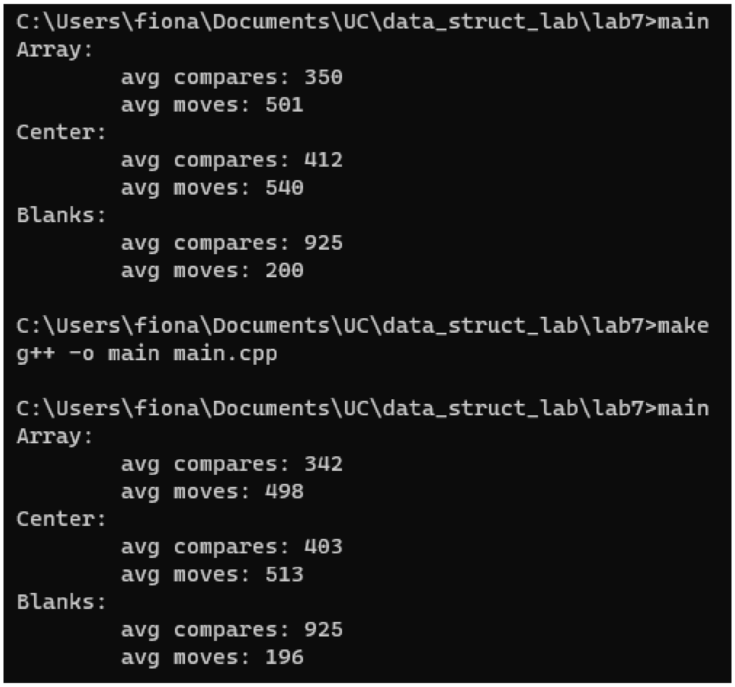
We believe that the Blanks class will have the least amount of move operations but may have more compare operations since it uses more compares in order to determine more efficient moves. The Center class will likely be more efficient than the original Array class. But when it comes to compare operations, the Array class might end up having the fewest.

Task 4 results and discussion

To test the three classes, we created a pointer array of integers and filled it with randomly generated numbers between 1 and 100. We run the program inside a for loop in the main function. We had each class count the number of compares and moves in a private int variable, which we get for each iteration and add to variables in the main function in order to display the averages at the end.

We considered running the whole program for the 100 tests from the command line, or using another similar method instead of just a for loop inside main, but we decided to go with this option because it is a lot simpler to get and calculate averages from inside the same function.

The results of our tests showed that while Blanks had the most compare operations, it also had far fewer move operations. Array and Center were closer in their ratio of average operations but Array tended to have fewer moves and compares in all the tests.

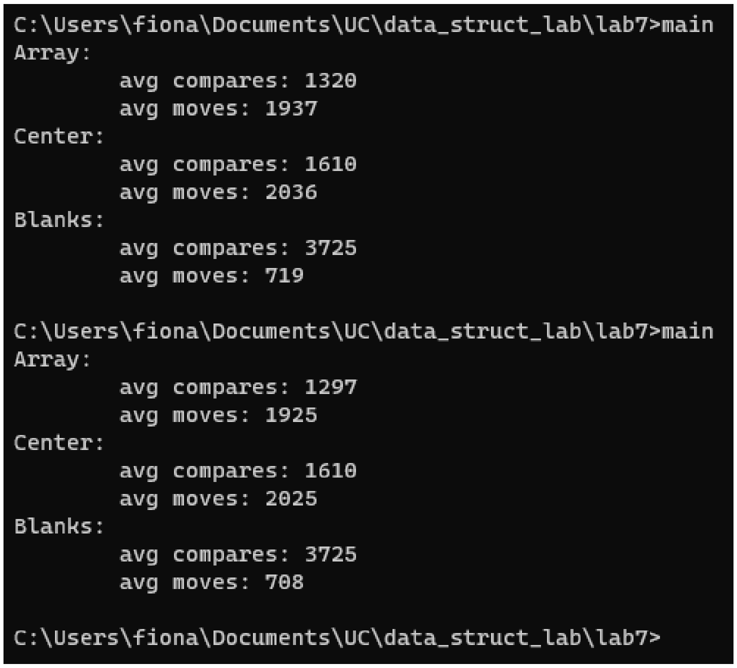


```
C:\Users\fiona\Documents\UC\data_struct_lab\lab7>main
Array:
        avg compares: 350
        avg moves: 501
Center:
        avg compares: 412
        avg moves: 540
Blanks:
        avg compares: 925
        avg moves: 200

C:\Users\fiona\Documents\UC\data_struct_lab\lab7>make
g++ -o main main.cpp

C:\Users\fiona\Documents\UC\data_struct_lab\lab7>main
Array:
        avg compares: 342
        avg moves: 498
Center:
        avg compares: 403
        avg moves: 513
Blanks:
        avg compares: 925
        avg moves: 196
```

(tests with 25 items)

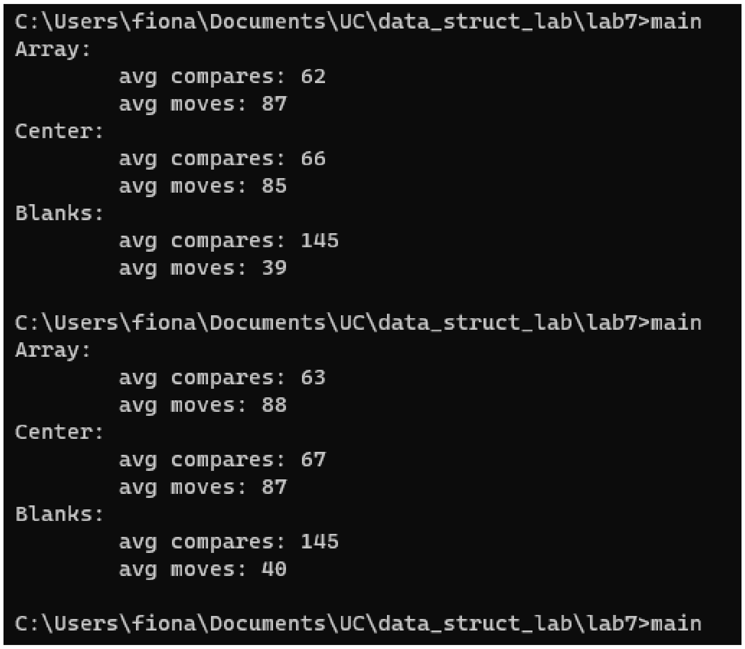


```
C:\Users\fiona\Documents\UC\data_struct_lab\lab7>main
Array:
        avg compares: 1320
        avg moves: 1937
Center:
        avg compares: 1610
        avg moves: 2036
Blanks:
        avg compares: 3725
        avg moves: 719

C:\Users\fiona\Documents\UC\data_struct_lab\lab7>main
Array:
        avg compares: 1297
        avg moves: 1925
Center:
        avg compares: 1610
        avg moves: 2025
Blanks:
        avg compares: 3725
        avg moves: 708

C:\Users\fiona\Documents\UC\data_struct_lab\lab7>
```

(tests with 50 items)



```
C:\Users\fiona\Documents\UC\data_struct_lab\lab7>main
Array:
        avg compares: 62
        avg moves: 87
Center:
        avg compares: 66
        avg moves: 85
Blanks:
        avg compares: 145
        avg moves: 39

C:\Users\fiona\Documents\UC\data_struct_lab\lab7>main
Array:
        avg compares: 63
        avg moves: 88
Center:
        avg compares: 67
        avg moves: 87
Blanks:
        avg compares: 145
        avg moves: 40

C:\Users\fiona\Documents\UC\data_struct_lab\lab7>main
```

(tests with 10 items)