Hunter Lavender

Dr. Devine

CS-111

3 May 2021

Algorithm Battle Ground

1. Which searching algorithm was the fastest for each data structure? Why?

The fastest searching algorithm is sequential search for our test conducted. The reason being is because for the given Binary Search to work we must have the data sorted. Naturally, I included the time to sort it as well. I used quick sort for this, because quick sort has generally been the fastest when running each experiment for each structure. This gives the best scenario, but the time to include this was needed to be noted, otherwise I feel it could give you a misconstrued view of the sequential search in comparison which finds it without it being sorted. (Far more impressive of a time and not every structure you are searching is going to sorted). Binary Search and Sequential search for Array List are very close to one another, and results vary on the data size. For temporary results they tie; however, this could easily change while I wait for the remaining test to run. In a link list the fastest is sequential without a doubt. Then moving on to doubly same case the sequential search is way faster than the other two.

1. Which sorting algorithm was the fastest for each data structure? Why?

For array List the fastest sorting algorithm is quick sort. This is because quick sort is dividing and conqueror sorter, and most of the time is going to be the best. Even in a Single Link List which are the worst structure to face in this scenario, quick sort makes this structure way more efficient in comparisons with the results, and Doubly Linked List quick sort blows the others away.

1. Which data structure was the fastest for each algorithm? Why?

Quick sort was the fastest for Array List, Linked List, Doubly Linked List. This is because the divide and conqueror averages are statistically going to be better. This is because even it does it slower a few times the law of averages really favors this method. On certain situations though some other ones might be better. That is why every scenario you have you must analyze and consider scenarios that could affect the speed of your data being sorted.

1. Were there any surprising results?

I remember when I first learned Linked List and thought the data structure was so cool. Then I learned about doubly linked list and realized linked list really was not nearly as cool as doubly linked list. This experiment solidified that thought. I loved how we went about doing this project, but I learned why not just use doubly linked list? I mean linked list is cool, but array list just makes more sense most of the time, and doubly is more efficient than the both of them generally speaking. So, if you wanted something like a linked list just use the doubly linked list. It is so much better all the way around and I honestly understand why. The way the sorting works with this structure is impressive. To see how they fly through the sorts makes it a clear winner which I expected, but you really learn how keeping track of the head and the tail make a MASSIVE difference. All around this has probably been my favorite project since I have been in college. I learned a lot this semester and furthered my passion for computer science! The sorting algorithms and figuring out how to make them more efficient has been a blast. Making my program more efficient (or at least learning how to after this project) is so much fun we really got everything we learned this semester and compared which one worked best in certain scenarios. Have a great summer!