Project 3

In this project, we are asked to implement 3 sorting styles in our own methods. For the insertion sort, we need to use a doubly linked list, while we can use either an array or a linked list for the merge sort and the quick sort. In order to use generics properly, we have to extend T as its super class in Comparable in order to make comparison between elements. A driver program is also needed to test all the functions implemented in the sorting classes.

Run time analysis

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best | Worst | Average |
| mergeSort() | O(n) | O(n) | O(n) |
| merge() | O(nlogn) | O(nlogn) | O(nlogn) |
| quickSort() | O(nlogn) | O(nlogn) | O(n2) |
| partition() | O(n) | O(n) | O(n) |
| insertionSort() | O(n2) | O(n) | O(n2) |

Feedback

This assignment gives me enough time to finish it. For the last two assignments, as along with this assignment, I did all the debug thing by myself. I kind of felt that this was why I didn’t have enough time for the last two projects. The debug just took me hours and hours, but for this project, I finished them on time, and I am going to learn how to use debug functions like eclipse. The difficulty for the project is fair, too. All the stuff that covered in class is related to the project.

1. Which algorithm performs the best when the data is already sorted or nearly so?

Insertion Sort

1. If reliable performance is important which sorting algorithm is best?

Merge Sort

1. What can cause Quick sort to perform poorly?

The pivot point is at the begin or the end, which does not divide the array into two pieces at all.