mergesort_np03cs4a220318

1. What are the most challenging aspects of the coursework task?

One challenging aspect of making this code was understanding the merge sort algorithm and implementing it correctly. Merge sort is a divide and conquer algorithm that involves dividing the input array into smaller subarrays, sorting the subarrays, and then merging the sorted subarrays back together.

And at first, I made the code in an array but after consulting my teacher, we were supposed to use arraylist and so another challenging aspect was converting the code from using an array to using an ArrayList. This involved replacing the array with an ArrayList and making sure to use the appropriate ArrayList methods (add and get instead of array indexing).

2. How did you go about completing the task?

To complete the task of making this code, I first familiarized myself with the merge sort algorithm. I read about how it works and how to implement it in code. Then, I converted the code from using an array to using an ArrayList. I encountered some challenges and bugs along the way, which required me to use my problem-solving skills to figure out what was going wrong. Overall, completing this task required a lot of focus and attention to detail. I learned about algorithms, data structures, and programming concepts, and developed my problem-solving skills in the process.

3. What have you learned over the course of completing this coursework task?

As I made this code by myself, I learned a lot of valuable skills and concepts. For example, I learned how to design and implement algorithms to solve problems, such as the merge sort algorithm. This involved breaking down a problem into smaller subproblems and using a logical approach to solve each subproblem and combine the solutions together.

I also learned about different data structures, such as arrays and ArrayLists, and how to use them effectively in my code. I learned about the trade-offs between different data structures and when to use each one.

In addition, I learned about programming concepts such as loops, conditionals, and function calls, as well as good programming practices like debugging, testing, and commenting.

As I worked on this code, I developed my problem-solving skills by debugging and troubleshooting issues that arose.