Project 1:

Name : Sumit Jain

UTD ID: sxj132730

Problem 1: Merge Sort

Java file : Sumit.java

Below are the comparison results of all 3 implementation of merge sort.

|  |  |  |  |
| --- | --- | --- | --- |
| Data Size | Dynamic Memo Allocation Algo | Using Auxillary Array with copying data | Alternate Merging Between Primary and Auxillary Array |
| 1 Million | 600-630 mili sec | 450-470 mili sec | 400-425 mili sec |
| 10 million | 6 secs | 3.6-4.2 sec | 2.3 sec |
| 100 million | 46-57 sec | 36-41 sec | 17-22 sec |

Program is created with menu for all 3 implementation.Enter ‘1’ for Dynamic Memo Allocation Algo, ‘2’ for Using Auxillary Array with copying data, ‘3’ for Alternate Merging Between Primary and Auxillary Array. Once this step is done, User has to enter the size of array which needs to be sorted. The program will created array with random numbers. Once the sorting is completed, the program will display the running time for the algorithm and menu options will be displayed. Enter ‘4’ for Exit

**Sample Output:**

D:\G\New folder>java -Xms512M -Xmx1524M Sumit

Enter Your Choice

1. Merger Sort Using Dynamic memory Allocation

2. Merger Sort Using Auxillary Array with Copying Data

3. Merger Sort Using Alternate Merging between Primary Array and Auxillary Array

4. Exit

2

Enter the Array Size :

10000000

ARRAY SORTED

It took 3885 milliseconds

Enter Your Choice

1. Merger Sort Using Dynamic memory Allocation

2. Merger Sort Using Auxillary Array with Copying Data

3. Merger Sort Using Alternate Merging between Primary Array and Auxillary Array

4. Exit

3

Enter the Array Size :

10000000

ARRAY SORTED

It took 2235 milliseconds

Enter Your Choice

1. Merger Sort Using Dynamic memory Allocation

2. Merger Sort Using Auxillary Array with Copying Data

3. Merger Sort Using Alternate Merging between Primary Array and Auxillary Array

4. Exit

4

D:\G\New folder>

**NOTE: Please run using** java -Xms512M -Xmx1524M Sumit , to ensure that program doesn’t run out heap memory space (in case of 100 million)