

Lab 02(E/11/133)

The problem of finding the odd person used few algorithms for sorting, changing data structures and more importantly to find the odd man.

Three classes are created.

- StudentData
- ExtractData
- OddMan

Student data are extracted using Jdbc driver. Then they were stored in an array list. To store the register number of a student and the personality number of that student together the class StudentData is created. In here we allow creating an object which has a double and string data fields. Then this array list is converted to an array. This particular operation has a time complexity of order n .

To sort this particular array a method called mergeSort is created. This sorting uses the merge sort algorithm which has a time complexity of $O(n\log(n))$.

The algorithm used to solve the problem is explained below graphically.

Let's assume we have this array.

53	134	123	45	54	187	97
----	-----	-----	----	----	-----	----



Sorting

45	53	54	97	123	134	187
----	----	----	----	-----	-----	-----



Creates another array called diff to store the differences of adjacent elements.

45	53	54	97	123	134	187
----	----	----	----	-----	-----	-----

53-45 54-53 97-54 123-97 134-123 187-134

8	1	43	26	11	53
---	---	----	----	----	----

Now every element in diff array compares against the adjacent element. If the 1st element is smaller than the next, the elements in original array which gives this value in diff array is nullified. So the resulting array is,

45	Null	Null	97	null	null	187
----	------	------	----	------	------	-----

Then again compares elements.

Finally the odd man of this array is 187.

This algorithm has a time complexity of order n.