CSSE230: Sorting Races

# Name(s): Yuankai Wang, Zishan Liu

# Part 1: Data(Using 3 million data)

Table 1 shows the runtimes of 6 sorts for at least 4 different types of arrays:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sort\array | Random order (w/ dup) | Random order (w/o dup) | Almost sorted | Almost sorted (reverse) |
| mergesort | 1174 milliseconds | 915 milliseconds | 113 milliseconds | 99 milliseconds |
| quicksort(Java) | 362 milliseconds | 248 milliseconds | 59 milliseconds | 112 milliseconds |
| BSTsort | 3496 milliseconds | 3082 milliseconds | 3102 milliseconds | 680 milliseconds |
| heapsort | 4946 milliseconds | 2271 milliseconds | 716 milliseconds | 733 milliseconds |
| quicksort(ours) | 371 milliseconds | 342 milliseconds | 58 milliseconds | 407 milliseconds |

# Part 2: Discussion

Include your discussion of the runtimes in Table 1, as described in the specification.

In general, the almost sorted array can be sorted faster than random sort. Quick sort is usually the faster sorting method. Merge sort is kind of stable. Its performance is acceptable in all arrays.

The quicksort we implemented has almost the same runtime as Java’s quicksort, except the almost reverse sorted array. In general, the result is what we expected. It shows that we sort in the right way. The reason for the difference in the reversed, we believe, is that java can pick a better pivot point than ours, and we use quicksort all the way through, but Java use insertion sort when the array size is small. It is faster with a small array size.

Tree sort and heap sort are obviously slower because they need more action. Heap sort has a significant improvement with they don’t have duplication. We think it is because heap sort sometimes change the places of two same elements, which is not necessary. Tree sort works much better when the list is reversed. We think it is because when the elements are added into and deleted from the tree, it will need less rotation. But the time use does not change much between random and almost sorted. We think it is because we input the data into another data structure, and tree still need lots of work even the array is almost sorted.