654 Advanced Computing Concepts
Assignment2 Report

I confirm that I will keep the content of this assignment confidential. I confirm that I have not received any unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 may be assigned for copied work. **Tengxiaoyao (Tab) Tu, #104518447**

Task 1. Use class Sort.java provided in class, the dual-pivot Quicksort of Java 8 (Arrays.sort), and RadixSort.java provided in class.

Task 2. Do the following for Mergesort, Quicksort, Heapsort and dual-pivot Quicksort:

a. Create 100,000 random keys (of type long) and sort them. Repeat this 100 times.

b. Compute the average CPU time taken to sort the keys for the four methods.

c. Comment on the results and compare them to the average-case complexities discussed in class.

Answer 2:

The result is the average in 100 times without the maximum and minimum values.
All results are record in String format, and transfer to an Excel file.

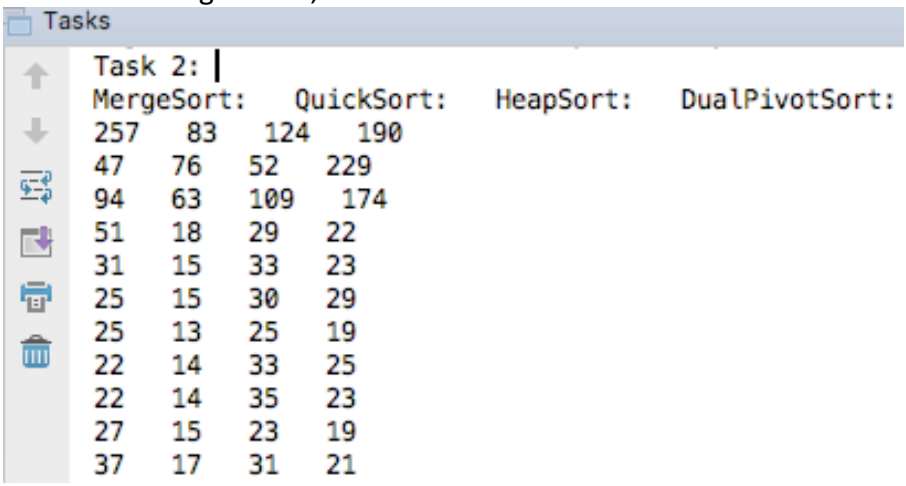

Figure 1. Screen Cut for Tasks 2

MergeSort:	QuickSort:	HeapSort:	DualPivotSort:
27.17ms	17.66ms	29.90ms	22.76ms

Table 1. Result for Tasks 2

As a result, Quick Sort is the fastest method in these for methods in sorting numbers, then Dual Pivot Quick Sort, Merge Sort and Heap Sort.

Task 3. Do the following for the four sorting methods of #2, and for Radix sort:

a. Create 100,000 random strings of length 4 and sort them using the five sorting methods.

b. Repeat (a) 10 times and compute the average CPU time that takes to sort the keys for the five methods.

c. Repeat (a) and (b) with strings of length 6, 8, 10.

d. Create a table with the results and compare the times with the average-case and worst-case complexities as studied in class.

Answer 3:

The result is the average in 100 times without the maximum and minimum values.
All results are record in String format, and transfer to an Excel file.

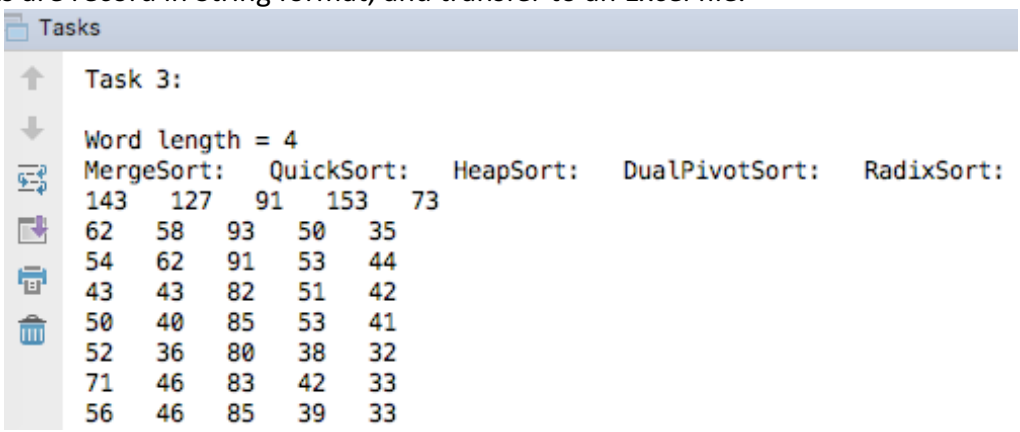

Figure 2. Screen Cut for Tasks 3

Word Length	MergeSort(ms):	QuickSort(ms):	HeapSort(ms):	DualPivotSort(ms):	RadixSort(ms):
4	60	48.625	88	49.125	36.875
6	59.375	44	93.875	49.25	58.25
8	55.875	42.625	90.375	49.875	87.375
10	59.625	47.25	103.25	50.5	115.625

Table 2. Result for Tasks 3

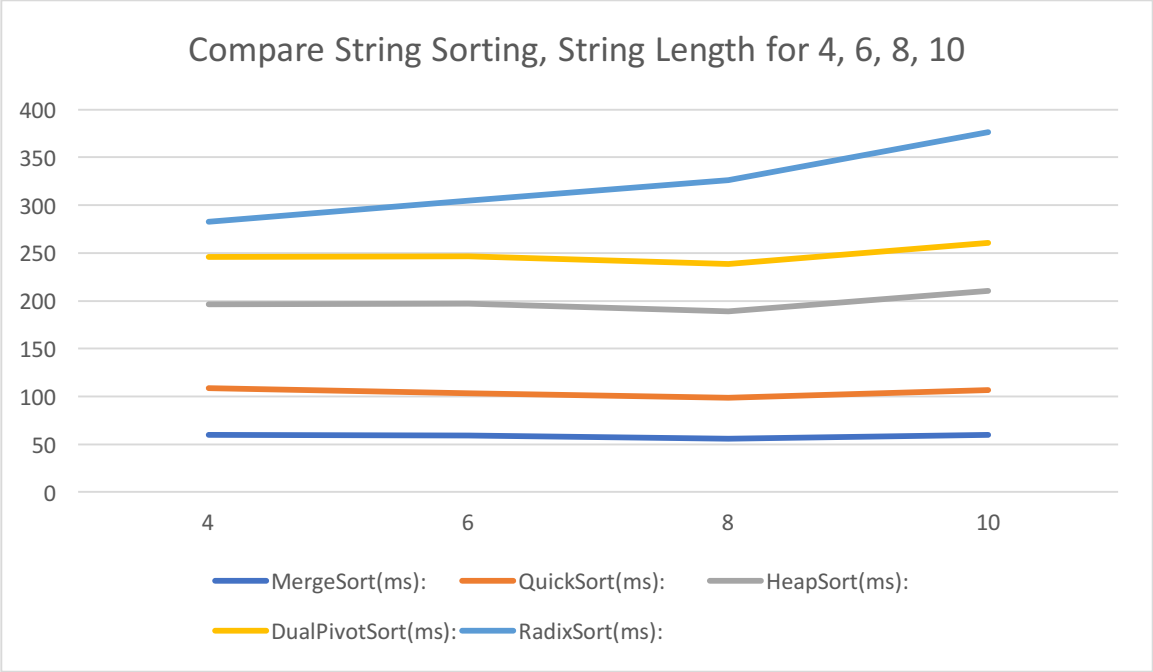

Figure 3. Compare String Sorting, String Length for 4, 6, 8, 10

In this chart, we can see Merge Sort is the fastest sorting method in these five algorithms in sorting strings. In addition, Radix Sort will become slow when the length of string grows.

4. Comment on: which sorting method will you use in your applications? in which case? Why?

Answer 4:

I will use Merge Sort in sorting strings in my applications since it is fast and less cost while big length, and Quick Sort for number sorting problems.

5. Use the edit distance (class Sequences.java) implementation provided in the source code.

a. Generate 1,000 pairs of random words of lengths 10, 20, 50 and 100.

b. Compute the edit distance for all words and find the average CPU time for each pair.

c. Compare the CPU times obtained for each word length with the running times of the edit distance algorithm.

Answer 5:

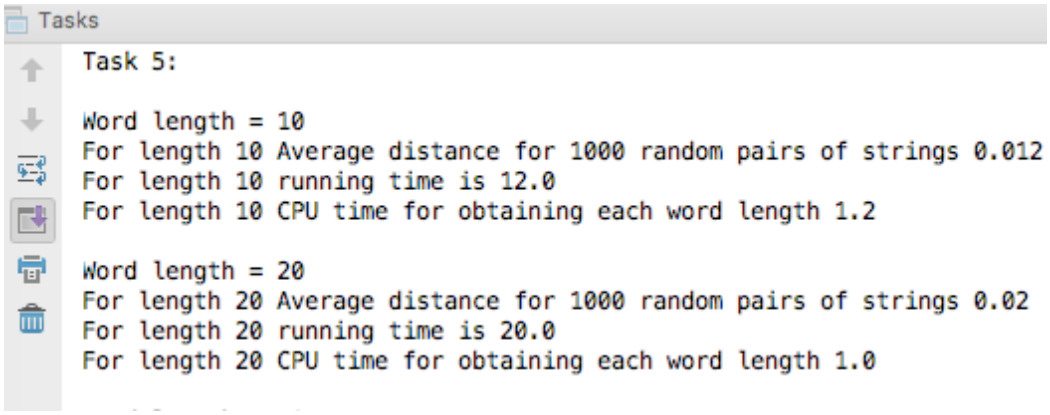

Figure 4. Screen Cut for Tasks 5

String Length	10	20	50	100
Average distance for 1000 random pairs of strings	0.012	0.02	0.04	0.104
running time	12	20	40	104
CPU time for obtaining each word length	1.2	1	0.8	1.04

Figure 2. Screen Cut for Tasks 3