

CS311 Yoshii - HW2 Part 2 – Sorting Analysis (based on week4 - week 5)

DUE: Week 6 Tuesday at the beginning of the class

TOTAL: 36 points Your score is:

Your NAME: Jer Cherng Law

Date Turned in: Sep 30 2014 Tuesday

Purpose: To demonstrate your understanding of analyzing searching and sorting algorithms.

A) Review Questions [1 pt per question = 5pts]

Your score is:

Type your answers here

- 1. Would you use Selection Sort or Insertion Sort? Why?**
Insertion sort. Because it has a better best case time complexity of $O(n)$ unlike Selection Sort which has $O(n^2)$.
- 2. Rika-Chu Sort corrects one inversion per comparison. What is its worst case number of comparisons? Why?**
- 3. What is the advantage of using Merge Sort over Quick Sort?**
It has worst case time complexity of $O(n \log n)$ unlike Quick Sort which has worst case time complexity of $O(n^2)$.
- 4. What is the disadvantage of using Merge Sort over Quick Sort?**
Quick sort uses less memory than Merge Sort.
- 5. Why is Radix sort unrelated to the $F(n) = O(n \log n)$ theorem?**
Because it doesn't make any comparisons.

B) Sort

230 123 324 10 23 56 (6 items)

using Insertion Sort and fill in the answers below.

[7pts]

Your score:

Start with pos 2 for X index:

Which items were shifted? 1

How many element comparisons until X is deposited back? 1

The resulting list is? 123 230 324 10 23 56

Start with pos 3 for X index:

Which items were shifted? 0

How many comparisons until X is deposited back? 2

The resulting list is? 230 123 324 10 23 56

Start with pos 4 for X index:

Which items were shifted? 3

How many comparisons until X is deposited back? 3

The resulting list is? 10 230 123 324 23 56

Start with pos 5 for X index:

Which items were shifted? 0

How many comparisons until X is deposited back? 1

The resulting list is? 123 230 324 10 23 56

Start with pos 6 for X index:

Which items were shifted? 0

How many comparisons until X is deposited back? 1

The final resulting sorted list is? 123 230 324 10 23 56

Q) Total number of comparisons was (add up the above): 8

Q) Give an example list for which you would have made the worst number of comparisons: 6 5 4 3 2 1

C) Using the Merge Sort algorithm, sort [8pts]

Your score:

8 5 6 3 9 2 1 7.

Fill in the []'s:

- 1. Break this up into:** [8 5 6 3] and [9 2 1 7]
- 2. Break these up into:** [8 5] and [6 3] [9 2] and [1 7]
- 3. Further Break these up into:** [8] and [5] [6] and [3] [9] and [2] [1] and [7]

Combine what and what? 8 and 5

Produce what? [5 8]

How many element comparisons for this part? 1

Combine what and what? 6 and 3

Produce what? [3 6]

How many comparisons for this part? 1

Combine what and what? 9 and 2

Produce what? [2 9]

How many comparisons? 1

Combine what and what? 1 and 7

Produce what? [1 7]

How many comparisons? 1

Combine what and what? [5 8] and [3 6]

Produce what? [3 5 6 8]

How many comparisons? 3

Combine what and what? [2 9] and [1 7]

Produce what? [1 2 7 9]

How many comparisons? 3

Combine what and what? [3 5 6 8] and [1 2 7 9]

Produce what? [1 2 3 5 6 7 8 9]

How many comparisons? 7

Q) Total number of comparisons was? 17

D) Sort 231 123 324 100 230 560 (6 items) using Radix Sort.

Hint: use 0-list, 1-list, 3-list, 4-list etc. [6pts]

Your score:

Pass1:

Show the sub-lists here based on the last char

'0'-bucket: 560 230 100

'1'-bucket: 231

'2'-bucket: -

'3'-bucket: 123

'4'-bucket: 324

'5'-bucket: -

'6'-bucket: -

Show the combined list

560 230 100 231 123 324

Pass2:

Show the sub-lists here based on the second char

'0'-bucket: 100

'1'-bucket: -

'2'-bucket: 123 324

'3'-bucket: 230 231

'4'-bucket: -

'5'-bucket: -

'6'-bucket: 560

Show the combined list

100 123 324 230 231 560

Pass3:

Show the sub-lists here based on the first char

'0'-bucket: -

'1'-bucket: 100 123

'2'-bucket: 230 231

'3'-bucket: 324

'4'-bucket: -

'5'-bucket: 560

'6'-bucket: -

Show the combined list

100 123 230 231 324 560

E) Program Merge Sort's Combine: [10 pts]

Your score:

Q) State of the program [2pts] <answer here>

- **Does your program compile without errors?** Yes
- **List any bugs you are aware of, or state “No bugs”:** No Bugs

Output test 1:

Comparing item from L1: 1 with item from L2: 4
Comparing item from L1: 2 with item from L2: 4
Comparing item from L1: 3 with item from L2: 4
Resulting vector is: 1 2 3 4 5 6

Output test 2:

Comparing item from L1: 1 with item from L2: 2
Comparing item from L1: 3 with item from L2: 2
Comparing item from L1: 2 with item from L2: 4
Comparing item from L1: 3 with item from L2: 4
Comparing item from L1: 5 with item from L2: 4
Comparing item from L1: 4 with item from L2: 6
Comparing item from L1: 5 with item from L2: 6
Resulting vector is: 1 2 3 4 5 6

Output test 3:

Comparing item from L1: 4 with item from L2: 1
Comparing item from L1: 1 with item from L2: 2
Comparing item from L1: 4 with item from L2: 2
Comparing item from L1: 2 with item from L2: 3
Comparing item from L1: 4 with item from L2: 3
Resulting vector is: 1 2 3 4 5 6