Mariya Eggensperger
CST 370, Spring 2017
Dr. Feiling Jia
Design/Analysis
of Algorithms

CST 370 Homework (Sorting)

1. (20 points) Sort the array of numbers 4, -3, 8, 1, 10 in ascending order using the selection sort algorithm. Show the state of the array after each iteration of the algorithm.

   ANSWER: The selection sort makes only one exchange for every pass through the list. See image and attached source code. A selection sort looks for the largest value as it makes a pass and, after completing the pass, places it in the proper location.

4	-3	8	1	10

*8 is the largest

4	-3	1	8	10

*4 is the largest

1	-3	4	8	10

*1 is the largest

FINAL

-3	1	4	8	10

*1 is the largest

2. (20 points) Sort the array of numbers 4, -3, 8, 1, 10 in ascending order using the bubble sort algorithm. Show the state of the array after each iteration of the algorithm.

ANSWER: The bubble sort makes multiple passes through a list. It compares adjacent items and exchanges those that are out of order. See image and source code attached.

Round 1 sort . . .

1st pass

----------> exchange

4	-3	8	1	10

2nd pass

----------> no exchange

-3	4	8	1	10

3d pass

----------> exchange

-3	4	8	1	10

4th pass

----------> no exchange

-3	4	1	8	10

Round 2 sort . . .

1st pass

----------> no exchange

-3	4	1	8	10

2nd pass

----------> exchange

-3	4	1	8	10

3d pass

----------> no exchange

-3	1	4	8	10

4th pass

----------> no exchange

-3	1	4	8	10

FINAL

-3	1	4	8	10

3. (25 points) Sort the array of numbers 10, 7, 3, 8, 1, 9, 0 in ascending order using the insertion sort algorithm. Show the state of the array after each iteration of the algorithm.

   ANSWER: Insertion sort maintains a sorted sublist in the lower positions of the list. Each new item is then inserted back into the previous sublist such that the sorted sublist is one item larger.

   *Assumes 10 is a sorted list of one item

   *Assumes 10 is a sorted list of one item

10	7	3	8	1	9	0

   *Inserted 7

7	10	3	8	1	9	0

   *Inserted 3

3	7	10	8	1	9	0

   *Inserted 8

3	7	8	10	1	9	0

   *Inserted 1

1	3	7	8	10	9	0

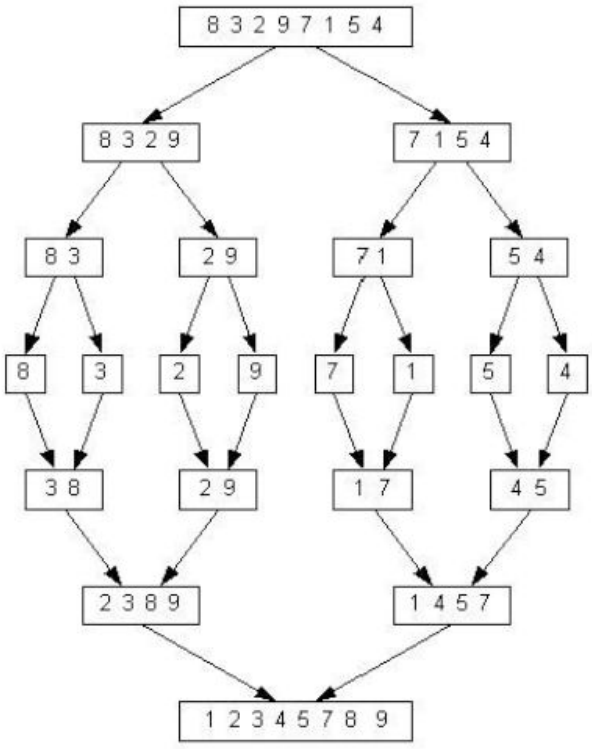
   *Inserted 9

1	3	7	8	9	10	0

*Inserted 0

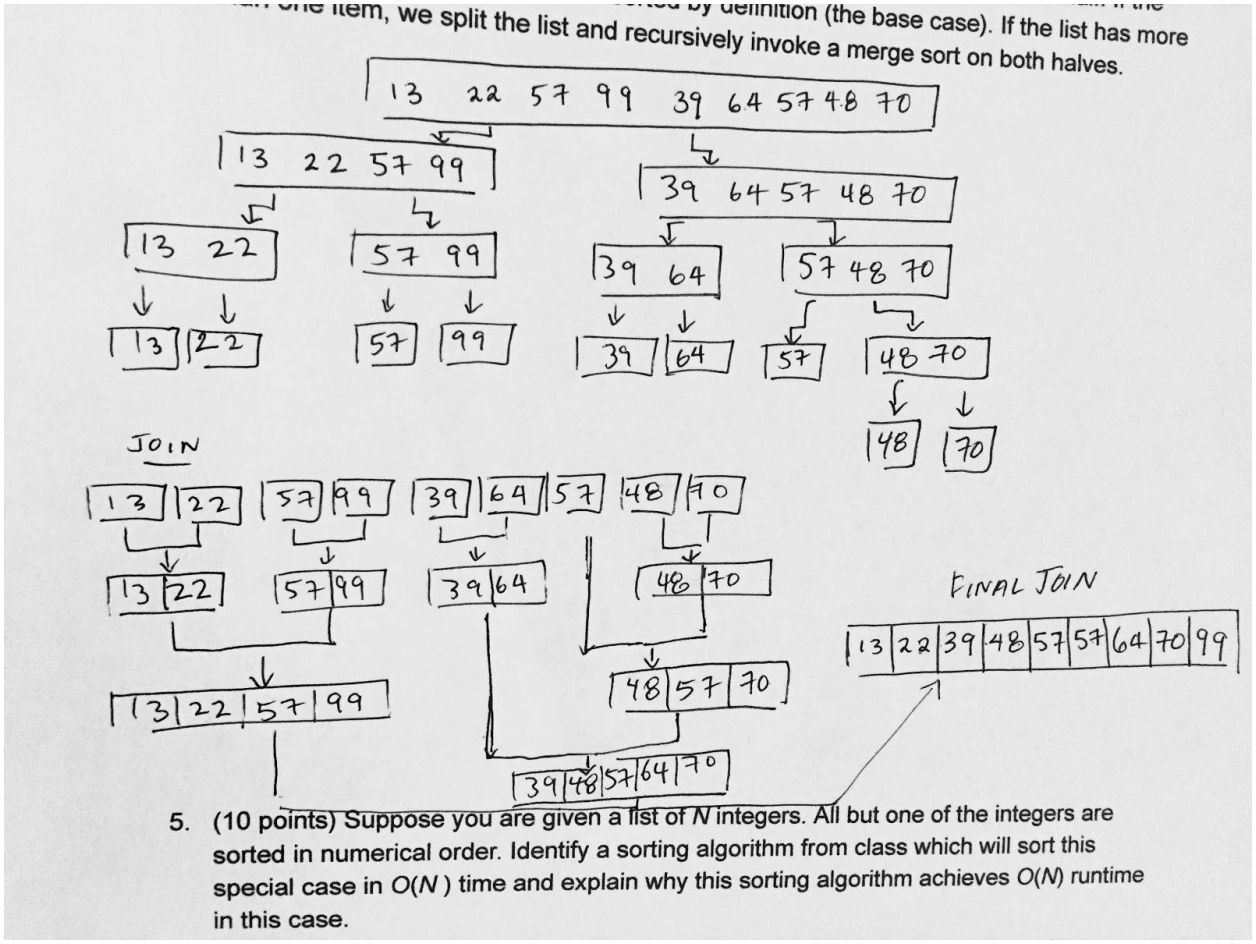
0	1	3	7	8	9	10

4. (25 points) Sort the array of numbers: 13, 22, 57, 99, 39, 64, 57, 48, 70 in ascending order using the merge sort algorithm. Show the state of various arrays after each iteration of the algorithm using the diagram similar to the one used in the supplemental materials, as shown below.



ANSWER: Merge sort is a recursive algorithm that continually splits a list in half. If the list is empty or has one item, it is sorted by definition (the base case). If the list has more than one item, we split the list and recursively invoke a merge sort on both halves.

(I had to write this one out myself in hand and then take a photo to input. It was taking too long to graphics design this).



5. (10 points) Suppose you are given a list of *N* integers. All but one of the integers are sorted in numerical order. Identify a sorting algorithm from class which will sort this special case in *O*(*N*) time and explain why this sorting algorithm achieves *O*(*N*) runtime in this case.

5. (10 points) Suppose you are given a list of *N* integers. All but one of the integers are sorted in numerical order. Identify a sorting algorithm from class which will sort this special case in *O*(*N*) time and explain why this sorting algorithm achieves *O*(*N*) runtime in this case.

Given a list of N integers, the counting sort assumes that each of the elements is an integer in the range 1 to *k*, for some integer *k*. When *k* = O(*n*), the Counting-sort runs in O(*n*) time. The basic idea of Counting sort is to determine, for each input elements *x*, the number of elements less than *x; for this reason, it would be a good solution to the out of order single element* and the information can be used to place directly into its correct position. See source code for implementation.