Michael Kohn
HW #5
CS570

2. For the following list: 9 0 11 10 5 8 6 7
show the sequence of changes using the
a: Bubble Sort Algorithm
b: Selection Sort Algorithm
c: Insertion Sort Algorithm
d: Merge Sort Algorithm

a: Bubble Sort Algorithm:
Repeatedly steps through the list to be sorted and compares each pair of adjacent items and swaps them if necessary. Keeps making passes until no swaps are needed.

0: 9 0 11 10 5 8 6 7
1: 0 9 11 10 5 8 6 7 swap 0 and 9
2: 0 9 10 11 5 8 6 7 skips 9,11 compare, swaps 11, 10
3: 0 9 10 5 11 8 6 7 swaps 11, 5
4: 0 9 10 5 8 11 6 7 swaps 11, 8
5: 0 9 10 5 8 6 11 7 swaps 11, 6
6: 0 9 10 5 8 6 7 11 swaps 11, 7
7: 0 9 5 10 8 6 7 11 returns to 0; 0,9 and 9,10 sorted, swaps 10,5
8: 0 9 5 8 10 6 7 11 swaps 10, 8
9: 0 9 5 8 6 10 7 11 swaps 10, 6
10: 0 9 5 8 6 7 10 11 swaps 10, 7
11: 0 5 9 8 6 7 10 11 returns to 0, 0,9 are sorted swaps 9, 5
12: 0 5 8 9 6 7 10 11 swaps 9, 8
13: 0 5 8 6 9 7 10 11 swaps 9, 6
14: 0 5 8 6 7 9 10 11 swaps 9, 7
15: 0 5 6 8 7 9 10 11 returns to 0, 0,5 are sorted, swaps 8, 6
16: 0 5 6 7 8 9 10 11 swaps 8, 7, no swaps needed
sorted!!

Michael Kohn
HW #5
CS570

2b: Selection Sort:
Find the smallest value in the list and switch value with the first position. Find the second smallest in the list and swap with the value in the second position. Continue until this pattern until the list is sorted.

0: 9 0 11 10 5 8 6 7	initial list	
1: 0 9 11 10 5 8 6 7	swaps	0, 9
2: 0 5 11 10 9 8 6 7	swaps	9, 5
3: 0 5 6 10 9 8 11 7	swaps	11, 6
4: 0 5 6 7 9 8 11 10	swaps	10, 7
5: 0 5 6 7 8 9 11 10	swaps	9, 8
6: 0 5 6 7 8 9 10 11	swaps	10, 11 - sorted!!

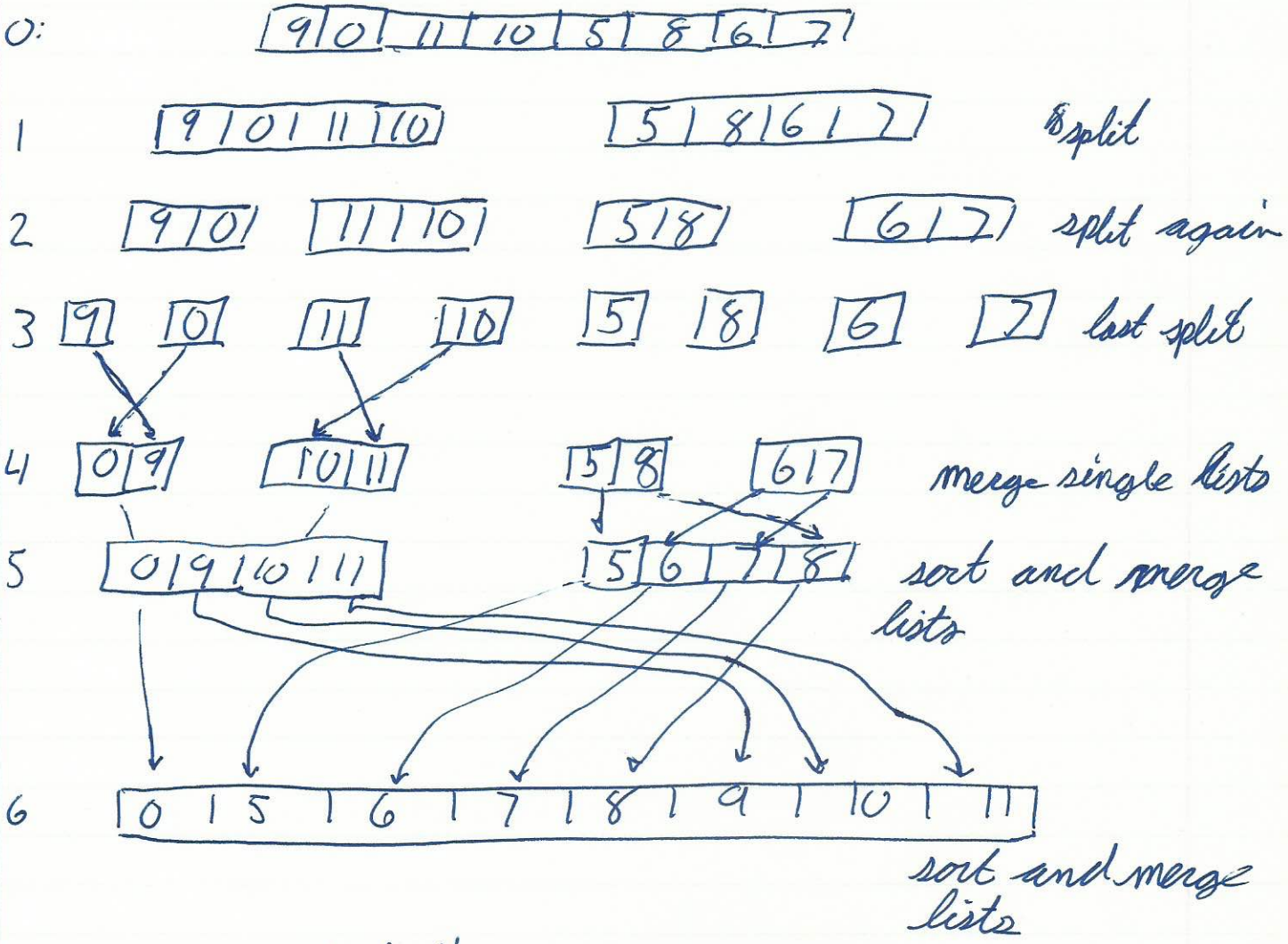
2c: Insertion Sort
Consider the first item of the list to be sorted. Insert second element, exchanging if necessary, Insert the third element in the appropriate place in relation to the first two. Continue until the list is sorted.

0: 9 0 11 10 5 8 6 7	initial list, 9 is sorted list
1: 0 9 11 10 5 8 6 7	add 0 to sorted list
2: 0 9 11 10 5 8 6 7	add 11 to sorted list
3: 0 9 10 11 5 8 6 7	add 10 to sorted list
4: 0 5 9 10 11 8 6 7	add 5 to sorted list
5: 0 5 8 9 10 11 6 7	add 8 to sorted list
6: 0 5 6 8 9 10 11 7	add 6 to sorted list
7: 0 5 6 7 8 9 10 11	add 7 to sorted list sorted!!

Michael Kohn
HW # 5
CS 570

2d: Merge Sort

Repeatedly breaks list into roughly equal pieces. This continues until each part is a list with a single element. Then merge the lists together sorting list if before merging. Continue sorting and merging until the list is the original size.



- sorted! -