Derek Acosta
10/3/16

COSC 160 Homework Sorting

For each show the portioning, swaps, and/or contents of auxiliary arrays. Basically I want to see the steps involved.

Sort the following sequence using 19, 4, 2, 9, 23, 18, 3, 20

a) Insertion sort

Iterations	19	4	2	9	23	18	3	20	input
1	**19**	4	2	9	23	18	3	20	Compare; 4 < 19; shift 19;insert 4
	4	19	2	9	23	18	3	20	2 < 19 & 2 < 4; shift 4 & 9; insert 2
2	**2**	4	9	19	23	18	3	20	19 > 9; correct position
3	2	4	9	**19**	23	18	3	20	23 > 19; correct position
4	2	4	9	19	**23**	18	3	20	18 < 23 & 18 < 19 & 18 > 9; insert 18
5	2	4	9	**18**	19	23	3	20	3 < 4 < 9 < 18 < 19 < 23 & 3>2; insert 3
6	2	**3**	4	9	18	19	23	20	19 < 20 < 23; insert 20
7	2	3	4	9	18	19	**20**	23	Sorted array

Note: red = subarrays

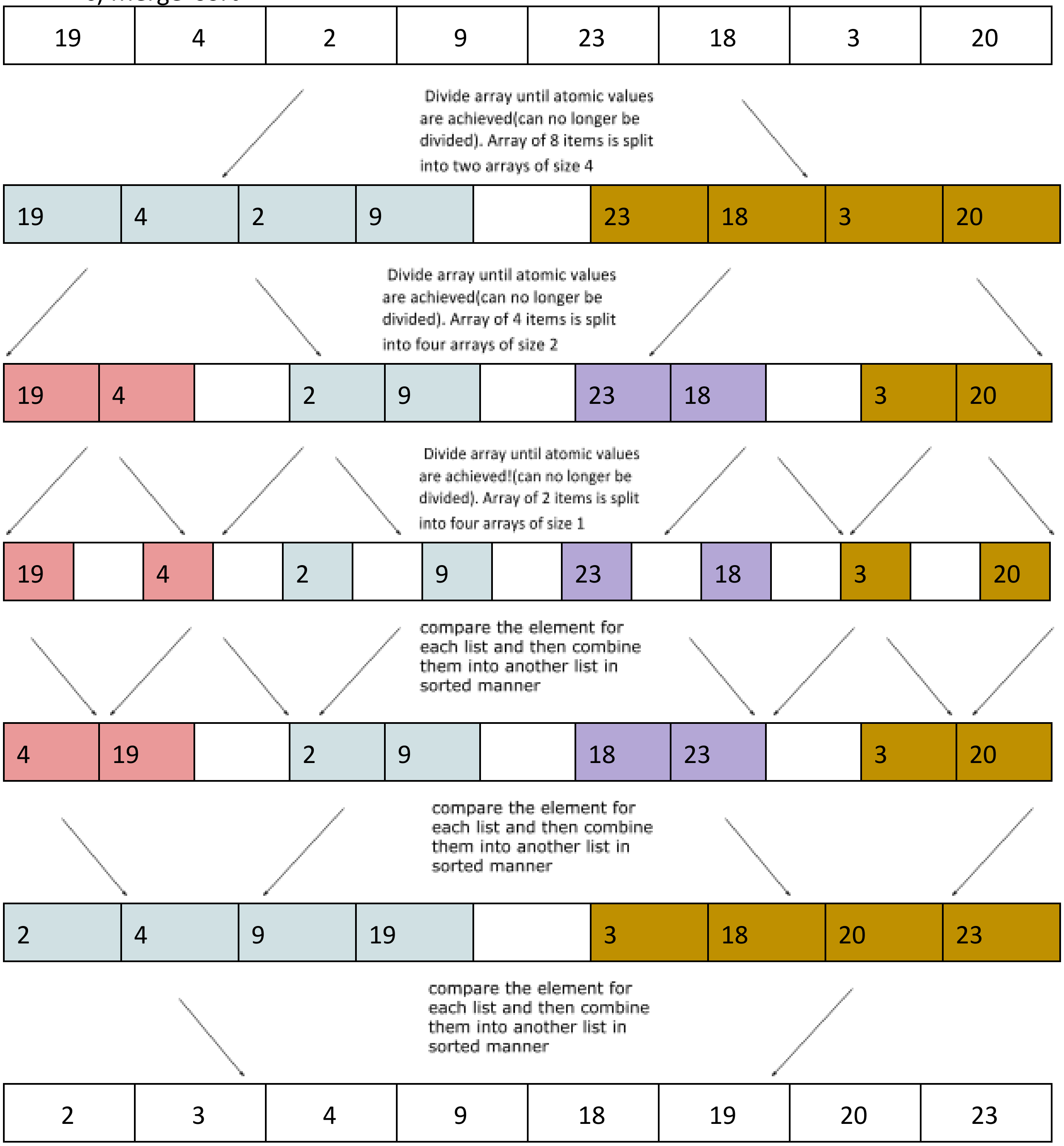
Derek Acosta
10/3/16

b) Quicksort (assume pivot is right-most key in array)

Index	19	4	2	9	23	18	3	20	Input; L = 19; R = 3; P = 20
	19	4	2	9	23	18	3	20	L move right until > P STOP; R move left if > P STOP; L = 23; R =3;
	19	4	2	9	3	18	23	20	If L & R don't match; Swap L & R; L = 3; R = 23
	19	4	2	9	3	18	23	20	L move right until > P STOP; R move left if > P;
	19	4	2	9	3	18	20	23	L >= R (they meet); swap index with pivot; new pivot
	19	4	2	9	3	18	20	23	Partitioning; PT1: L = 19; R = 3; P = 18; PT2: L = 23; R=23; P= 23; sorted!; acts as own partition; exclude original Pivot
	19	4	2	9	3	18	20	23	L move right until > P STOP; R move left if > P STOP; L = 19; R = 3;
	3	4	2	9	19	18	20	23	If L & R don't match; Swap L & R; L = 3; R = 19
	3	4	2	9	19	18	20	23	L move right until > P STOP; R move left if > P;
	3	4	2	9	18	19	20	23	L >= R (they meet); swap index with pivot; new pivot
	3	4	2	9	18	19	20	23	Partitioning; PT3: L = 3; R = 2; P = 9; PT4: L = 19; R=19; P= 19; sorted!; acts as own partition; exclude original Pivot
	3	4	2	9	18	19	20	23	L move right until > P STOP; R move left if > P STOP; L = 4; R = 2; because there are no number > P in array P act as own partition. PT5: L = 3, R =4, P=2
	4	3	2	9	18	19	20	23	If L & R don't match; Swap L & R;
	4	3	2	9	18	19	20	23	L move right until > P STOP; R move left if > P;
	2	3	4	9	18	19	20	23	L >= R (they meet); swap index with pivot; new pivot
	2	3	4	9	18	19	20	23	Partitioning; PT3: L = 3; R = 4; P = 2; P become partition
	2	3	4	9	18	19	20	23	L = 3; R = 3; P=4; do nothing; P becomes partition
	2	3	4	9	18	19	20	23	P = 3; L =3; R=3; becomes own partition
	2	3	4	9	18	19	20	23	Sorted!

Derek Acosta
10/3/16

c) Merge sort



19	4	2	9	23	18	3	20

Divide array until atomic values are achieved(can no longer be divided). Array of 8 items is split into two arrays of size 4

19	4	2	9		23	18	3	20

Divide array until atomic values are achieved(can no longer be divided). Array of 4 items is split into four arrays of size 2

19	4		2	9		23	18		3	20

Divide array until atomic values are achieved!(can no longer be divided). Array of 2 items is split into four arrays of size 1

19		4		2		9		23		18		3		20

compare the element for each list and then combine them into another list in sorted manner

4	19		2	9		18	23		3	20

compare the element for each list and then combine them into another list in sorted manner

2	4	9	19		3	18	20	23

compare the element for each list and then combine them into another list in sorted manner

2	3	4	9	18	19	20	23

Derek Acosta
10/3/16

d) Counting sort

Input:

19	4	2	9	23	18	3	20

1) Take a count array to store the count of each unique object.

Index & count:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
		1	1	1					1									1	1	1			1		

2) Modify the count array such that each element at each index stores the sum of previous counts.

Index & count:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	0	1	2	3	3	3	3	3	4	4	4	4	4	4	4	4	4	5	6	7	7	7	8	8	8
		0	1	2					3									4	5	6			7		

3) Output each object from the input sequence followed by decreasing its count by 1.

Since there are 8 input create an array with 8 places

1	2	3	4	5	6	7	8
2	3	4	9	18	19	20	23

e) Radix sort (base 10)

Organize list in ascending order from least to most significant digit

0 Pass: prefix zeros	1st Pass/ 1st Place	2nd Pass/2nd
19	20	02
04	02	03
02	23	04
09	03	09
23	04	18
18	18	19
03	19	20
20	09	23

Derek Acosta
10/3/16