CSE 373 HW6 P2 Writeup

Stephen Hung

1. Using Quick Sort, with the median pivot rule (pick the median of: data[lo], data[hi - 1], and data[(hi + lo) / 2]), sort the following list of numbers. Show your work by drawing the tree of partitions and pivots (as seen in the lecture slides) with the partition rules discussed in lecture (swapping the pivot to index lo and doing swaps to complete the partitions). Apply a cutoff of 3 elements and sort with any sorting method.

data =	5	7	9	1	3	4	6	8	2

   The three values to choose the pivot median from are:

   - data[lo] = data[0] = 5
   - data[(hi+lo)/2] = data[4] = 3
   - data[hi-1] = data[9 − 1] = 2
   - The median pivot is therefore the value at data[4] which is 3

   First swap the pivot with index 0

3	7	9	1	5	4	6	8	2
↑	↑							↑
pivot	left pointer							right pointer

   Left Pointer: $7 > 3$ so stop pointer

   Right Pointer: $2 < 3$ so stop pointer

   Since both left and right are stopped, swap the two pointers and then move the left and right pointer forward/backward.

3	2	9	1	5	4	6	8	7
↑		↑					↑	
pivot		left pointer					right pointer	

   Left Pointer: $9 > 3$ so stop pointer

   Right Pointer: $8 > 3$ so keep moving pointer.

3	2	9	1	5	4	6	8	7
↑		↑				↑		
pivot		left pointer				right pointer		

   Left Pointer: Still stopped at 9.

   Right Pointer: $6 > 3$ so keep moving pointer.

3	2	9	1	5	4	6	8	7
↑		↑			↑			
pivot		left pointer			right pointer			

   Left Pointer: Still stopped at 9.

Right Pointer: $4 > 3$ so keep moving pointer.

3	2	9	1	5	4	6	8	7
↑		↑		↑				
pivot		left pointer		right pointer				

Left Pointer: Still stopped at 9.

Right Pointer: $5 > 3$ so keep moving pointer.

3	2	9	1	5	4	6	8	7
↑		↑	↑					
pivot		left pointer	right pointer					

Left Pointer: Still stopped at 9.

Right Pointer: $1 < 3$ so stop pointer.

Since both pointers are stopped, swap the pointers and then move right pointer first.

3	2	1	9	5	4	6	8	7
↑		↑						
pivot		left pointer and right pointer						

Since the pointers are pointing at the same index, stop the sort and swap with pivot.

1	2	3	9	5	4	6	8	7
		↑						
		pivot						

Partition the array.

1	2	3	9	5	4	6	8	7

The left array | 1 | 2 | is below 3 elements and is thus sorted. Since it is already sorted it stays in the same order.

The right arrow is above 3 elements and thus goes through another round of quick sort. The three values to choose the pivot median from are

- data[lo] = data[0] = 9
- data[(hi+lo)/2] = data[3] = 6
- data[hi-1] = data[$6 - 1$] = 7

Thus the median pivot is data[5] = 7. Swap the element at index 0 with the pivot.

7	5	4	6	8	9
↑	↑				↑
pivot	left pointer				right pointer

Left Pointer: $5 < 7$, keep the pointer moving.

Right Pointer: $9 > 7$, keep the pointer moving.

7	5	4	6	8	9
↑		↑		↑	
pivot		left pointer		right pointer	

Left Pointer: $4 < 7$, keep the pointer moving.

Right Pointer: $8 > 7$, keep the pointer moving.

7	5	4	6	8	9
↑			↑		
pivot			left and right pointer		

Since the pointers are pointing at the same index, stop the sort and swap with pivot.

6	5	4	7	8	9
			↑		
			pivot		

Partition the array.

6	5	4	7	8	9

Since all partitions are ≤ 3 elements, sort all partitions.

6	5	4	7	8	9

↓

4	5	6	7	8	9

Combine the partitions

4	5	6	7	8	9

↓

4	5	6	7	8	9

The partition from the first quick sort now looks like this:

1	2	3	4	5	6	7	8	9

Combine these partitions

1	2	3	4	5	6	7	8	9

↓

1	2	3	4	5	6	7	8	9

Since this is the last partition, the quick sort is complete.

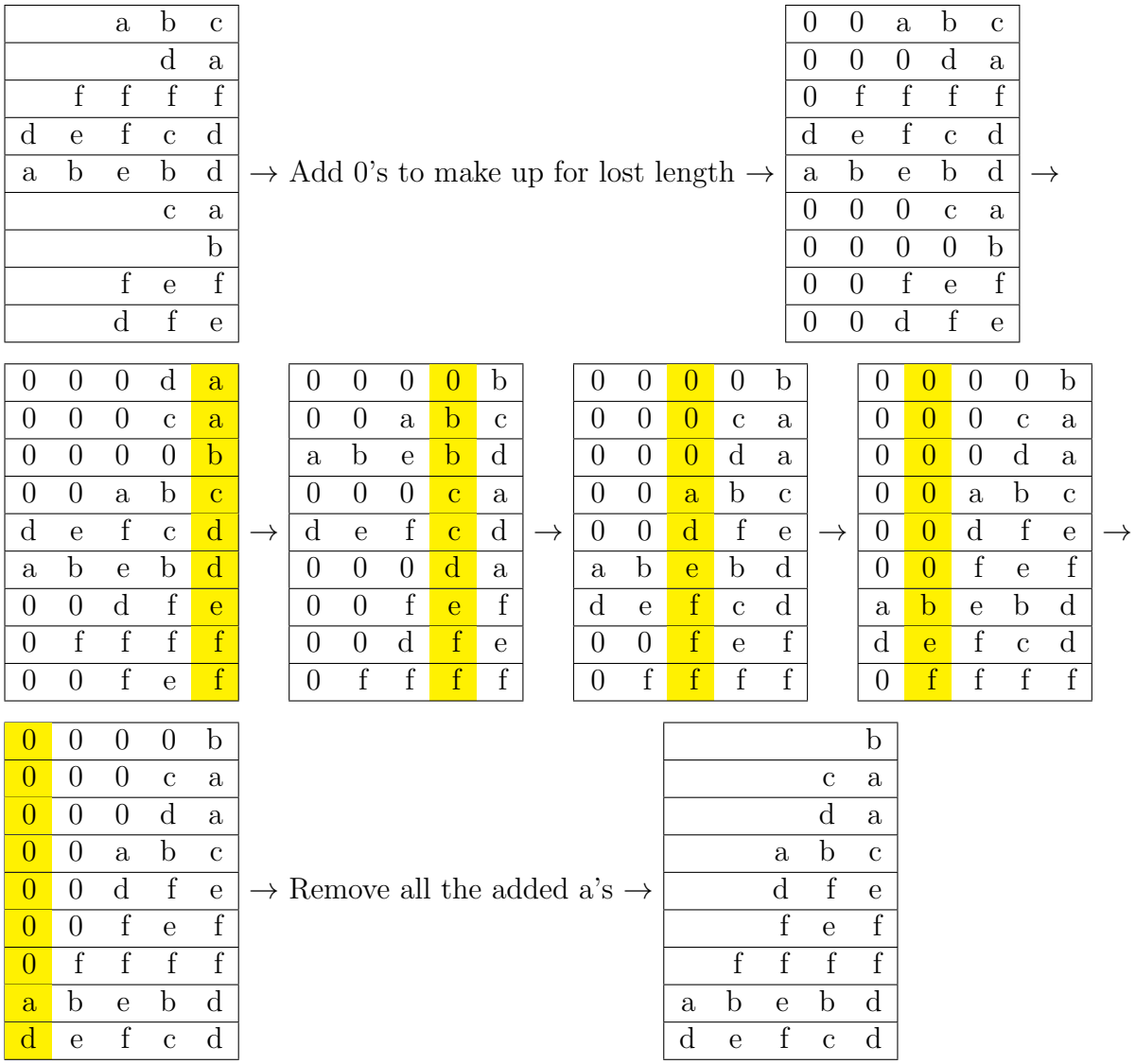
2. Using Radix Sort with a radix of 6 (letters: a, b, c, d, e, f) to alphabetically sort the following strings, draw contents of each bucket at the end of each radix 'digit' iteration pass.

Strings =

abc	da	ffff	defcd	abebd	ca	b	fef	dfe

For strings that have a length less than 5, a 0 can be added to their beginning to make up for the lost length.

Yellow indicates the column that is sorted at that step.


→ Add 0's to make up for lost length →
→
→
→
→
→
→ Remove all the added a's →