Lecture 15
Sorting
Exercises ANS

Department of Computer Science
Hofstra University

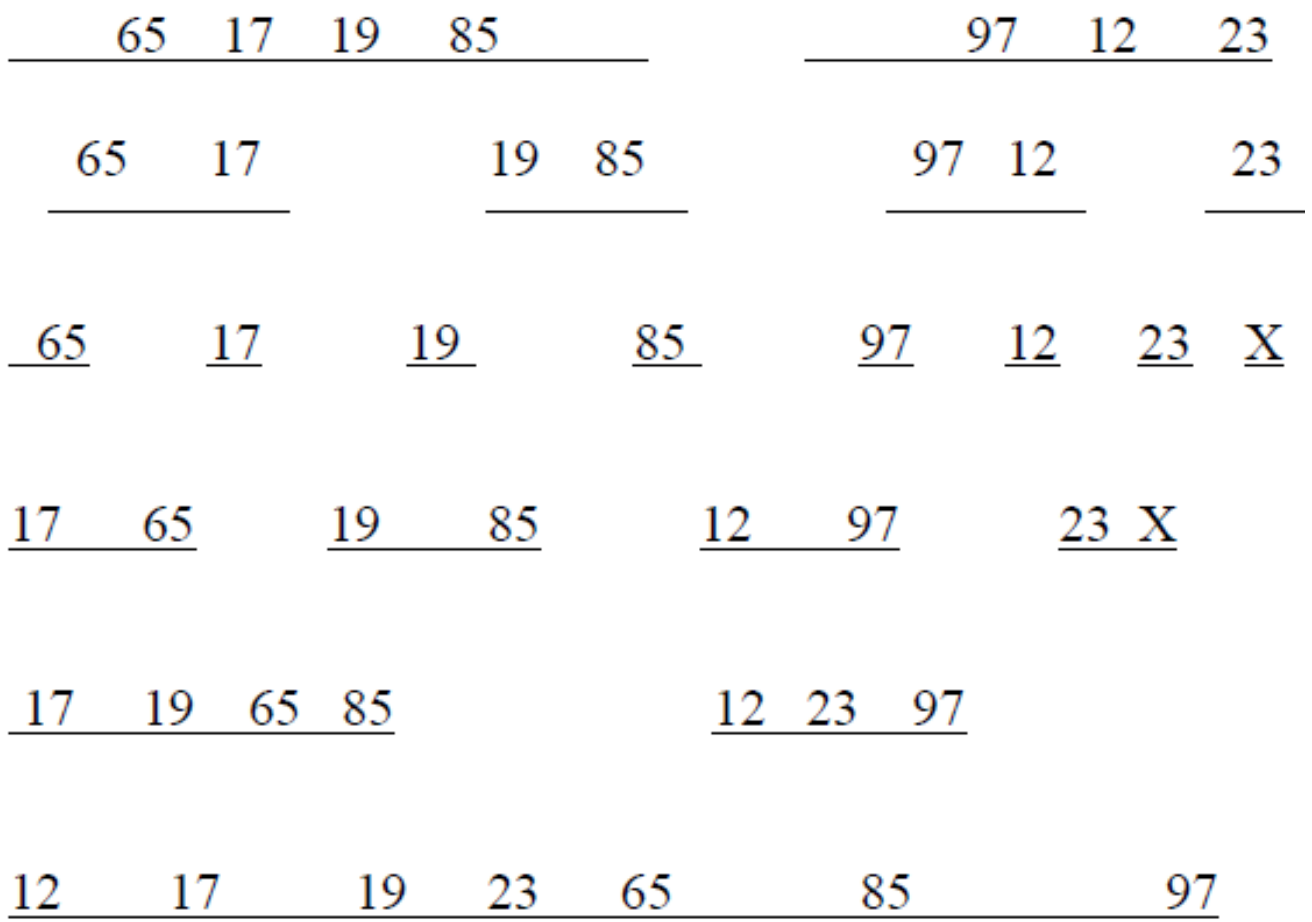
Acknowledgement: Lecture slides based on UofW Course on Data Structures

Q. Merge Sort ANS

Sort this list of numbers using mergesort. Show the split into sublists, then show the merge steps. When there are an odd number of elements in a list, make the left sublist larger. Put an ‘X’ on any sublist you don’t use.

65 17 19 85 97 12 23

ANS: fig to the right



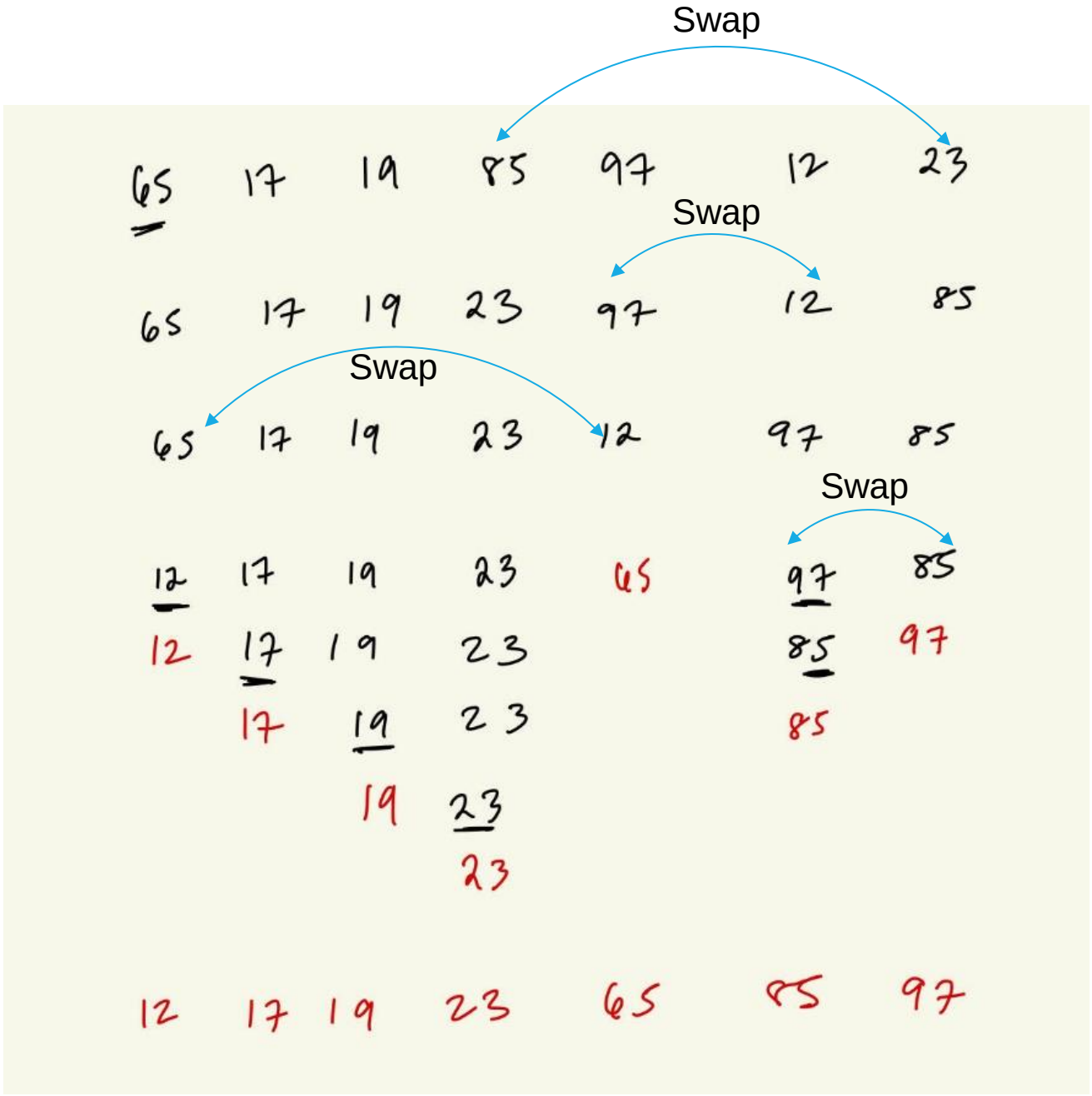
Q. Quick Sort ANS

Sort this array of numbers with **In-Place QuickSort** into ascending order, using the first number of each subarray as the pivot. Show the intermediate subarrays at each step, enclosing the pivot at each step with parentheses. Draw the corresponding Binary Search Tree and give the final sorted array.

65 17 19 85 97 12 23

ANS: fig to the right.

Please refer to “Quick Sort (v2: In-Place) Example I”, “Quick Sort (v2: In-Place) Example II” for how to perform the step-by-step swapping of elements for each pivot at each step. You must do this step-by-step swapping process to reach the correct answer, e.g., if you simply put (17, 19, 12, 23) to the left of pivot 65, and (85, 97) to the right, it is not correct according to our QuickSort algorithm, and you will get a different Binary Search Tree at the end (although the final result will still be sorted.)



Q. Quick Sort ANS

(Please show the detailed process as in the previous slide of swapping each pair of elements.)

12 17 19 23 (65) 97 85

Left part:

(12) 17 19 23

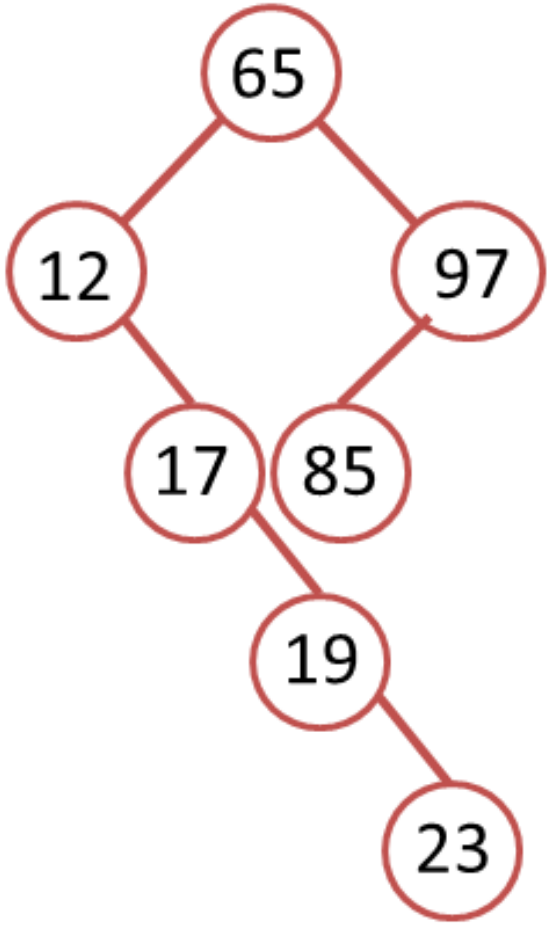
(17) 19 23

(19) 23

Right part:

85 (97)

Sorted list: 12 17 19 23 65 85 97



Corresponding BST

Q. Radix Sort ANS

Sort this array of numbers with Radix sort, with radix of 10, into ascending order. Show the intermediate results after each pass.

65 17 19 85 97 12 23

ANS: fig to the right

170	45	75	90	802	24	2	66

After 1st pass (sorting by the last digit)

170	90	802	2	24	45	75	66

After 2nd pass (sorting by the 2nd to last digit)

802	2	24	45	66	170	75	90

After 3rd pass (sorting by the first digit)

2	24	45	66	75	90	170	802

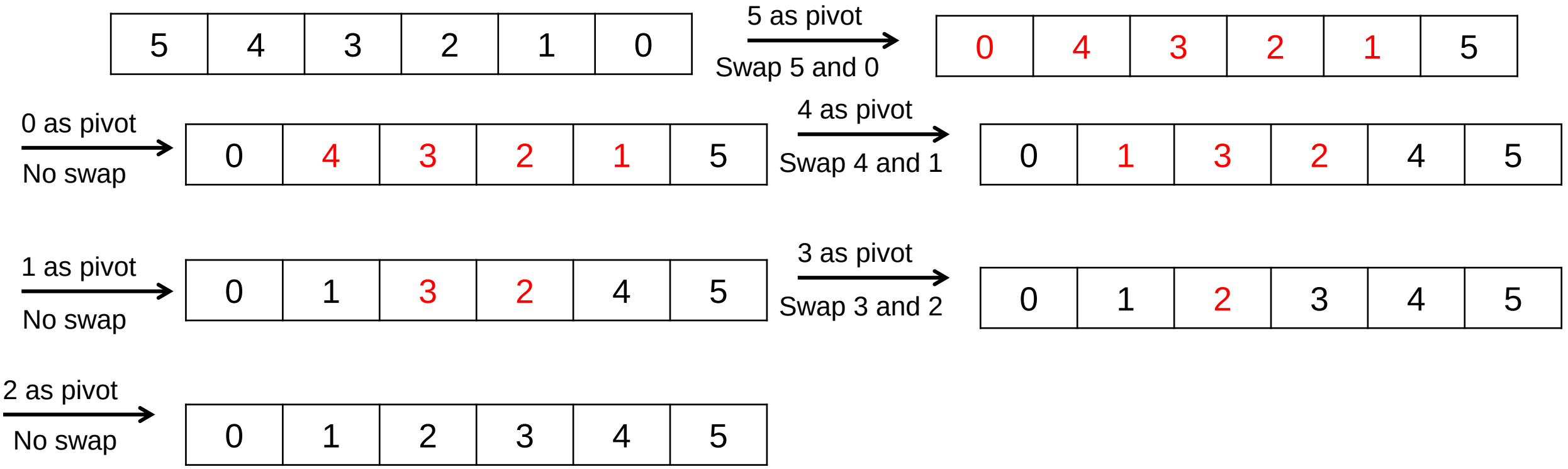
Q. Quick Sort

Sort this array of numbers with In-Place QuickSort into ascending order, using the first number of each subarray as the pivot. Show the intermediate subarrays at each step, enclosing the pivot at each step with parentheses. Draw the corresponding Binary Search Tree and give the final sorted array.

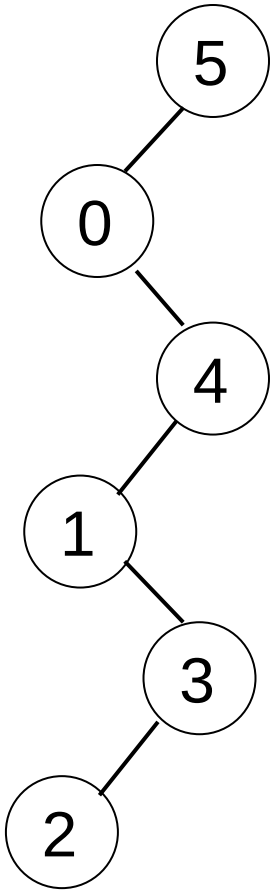
5, 4, 3, 2, 1, 0

Q. Quick Sort ANS

- Red numbers indicate remaining subarray to be sorted. In this example, every time the pivot leaves either left or right subarray as empty, so only one subarray is left after each step.



Q. Quick Sort ANS



Corresponding BST