Ch. 7.5-7.6 Sorting 習題

Q1

- Please show step-by-step Merge Sort to sort the following array and check whether this algorithm is stable. Array: [26 17 93 31 55 20 44 77 1 63].
- Please implement the iterative Merge Sort.

Ans 1: Ref: https://www.geeksforgeeks.org/iterative-merge-sort/

[26 17 93 31 55 20 44 77 1 63]

[26 17 93 31 55] [20 44 77 1 63]

[26 17] [93 31 55] | [20 44] [77 1 63]

[26] [17] [93] [31 55] | [20] [44] [77] [1 63]

[26] [17] [93] [31] [55] | [20] [44] [77] [1] [63]

[17 26] [31 93] [55] | [20 44] [1 77] [63]

[17 26 31 93] [55] | [1 20 44 77] [63]

[17 26 31 55 93] |[1 20 44 63 77]

[1 17 20 26 31 44 55 63 77 93]

Q2

- **Please complete the following table. You should give the time complexity in Big-O and explain the result.**

	Best case	Worst case
Insertion Sort		
Quick Sort		
Merge Sort		

Ans 2:

	Best case	Worst case
Insertion Sort	(n-1)* 1 = O(n)	1+2+…+(n-1) = $O(n^2)$
Quick Sort	O(n*logn)	$O(n^2)$
Merge Sort	O(n*logn)	O(n*logn)

- You can try to “feel” why
- To solve T(n) function: Use “substitution method” or “recursion tree”
- Or use “Master Theorem”:
  - Quick Sort:
    - Best case: $T(n) = 2T(n/2) + \theta(n) \Rightarrow O(n*\log n$
    - Worst case: $T(n) = T(n-1) + T(0) + \theta(n) = c(n-1) + c(n-2) \ldots\ldots + c = c(n-1)(n)/2 = O(n^2)$
  - Merge Sort:
    - For both Best case and Worst case: $T(n) = 2T(n/2) + \theta(n) \Rightarrow O(n*\log n)$

Master Theorem:

https://zh.wikipedia.org/wiki/%E4%B8%BB%E5%AE%9A%E7%90%86

Q3

- Write the status of the list (12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18) at the end of each phase of *MergeSort* (Program 7.9) (the “iterative” Merge Sort).

Ans 3:

- [2 12] [16 30] [8 28] [4 10] [6 20] [18]
- [2 12 16 30] [4 8 10 28] [6 18 20]
- [2 4 8 10 12 16 28 30] [6 18 20]
- [2 4 6 8 10 12 16 18 20 28 30]

Q4

- Suppose that we use Program 7.11 to obtain a Merge Sort function. Is the resulting function a stable sort? (the “recursive” Merge Sort)

Ans 4:

Assume merge two subarray:

- **[20 30a 40 50] [30b 60 70 90]**

-> [20 30a 30b .....]

Yes, merge sort is a stable sort.

Q5

- Write the status of the list (12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18) at the end of the first **for** loop as well as at the end of each iteration of the second **for** loop of *HeapSort* (Program 7.14).

Ans 5:

- max-heap tree: 30 20 28 12 18 16 4 10 2 6 8
- Each iteration of second loop:
  - Remove-heap tree: 28 20 16 12 18 8 4 10 2 6 [30]
  - Remove-heap tree: 20 18 16 12 6 8 4 10 2 [28 30]
  - Remove-heap tree: 18 12 16 10 6 8 4 2 [20 28 30]
  - Remove-heap tree: 16 12 8 10 6 2 4 [18 20 28 30]
  - Remove-heap tree: 12 10 8 4 6 2 [16 18 20 28 30]
  - Remove-heap tree: 10 6 8 4 2 [12 16 18 20 28 30]
  - Remove-heap tree: 8 6 2 4 [10 12 16 18 20 28 30]
  - Remove-heap tree: 6 4 2 [8 10 12 16 18 20 28 30]
  - Remove-heap tree: 4 2 [6 8 10 12 16 18 20 28 30]
  - Remove-heap tree: 2 [4 6 8 10 12 16 18 20 28 30]
  - Remove-heap tree: [2 4 6 8 10 12 16 18 20 28 30]