Sorting Worksheet

# Review Questions

1. Suppose we have a O(n) time algorithm that finds median of an unsorted array. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. What will be the worst case time complexity of this modified QuickSort?

if all elements in an array have the same value, the time complexity will be O(n^2).

1. What does it mean for a sorting algorithm to be stable?

if elements with the same key maintain the same relative order after sorting, then this sorting algorithm is stable.

1. Which of the following is not a stable sorting algorithm? C
   1. Insertion Sort
   2. Bubble Sort
   3. Quick Sort
   4. Merge Sort
2. What does it mean for a sorting algorithm to be in place?

in- place means that sorting algorithm does not need an extra space to store an output and it stores the output in the same memory that contains the input data before.

1. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: **[2, 5, 1, 7, 9, 12, 11]** What are all the possible values that could have been the pivot?

[7, 9]

1. You have to sort 1 GB of data with only 100 MB of available main memory. Which sorting technique will be most appropriate?

I found the answer from GeeksforGeeks

we would choose Merge Sort.

(1) Divide the data into 10 groups each of size 100MB.

(2) Sort each group and write them to disk

(3) Load 10 times from each group into main memory.

(4) Output the smallest item from the main memory to disk. Load the next item from the group whose item was chosen.

(5)Loop step (4) until all items are outputted

if we use this method, the time complexity is 10N\*log(N/10) + O(N)

Question: why don’t we choose quick sort? Just because we need to take the advantage of the main memory, so we choose the MergeSort.

1. Which of the following sorting algorithms has the fastest worst-case time complexity? C
   1. Selection sort
   2. Insertion Sort
   3. Quicksort
   4. Mergesort

Use the following code snippet for question 8 and 9

def sort(arr):

N = len(arr);

for i in range(N): #outer loop

for j in range(N-1, i, -1):

if arr[j] < arr[j-1]:

#swaps two elements

arr[j], arr[j-1] = arr[j-1], arr[j]

1. Which of the following is true about this sorting algorithm? A , B
   1. The sorting algorithm is stable.
   2. The sorting algorithm is in-place.
2. Which of these invariants does the above code satisfy at the end of each outer i loop? Select all that apply. A,B,D
   1. Entries arr[0] through arr[i] are in sorted order.
   2. Entries arr[0] through arr[i] contain the smallest keys in the entire array.
   3. Entries arr[N-i+1] through arr[N-1] are in sorted order.
   4. Entries arr[N-i+1] through arr[N-1] contain the largest keys in the entire array

# Tips for Solving Leetcode Problems

* Spend only 45 minutes trying to devise a solution to each problem (everything except coding)
* Please reference our [guide on how to approach practice exercises in Teachable](https://codebreakers1.teachable.com/courses/codebreakers-training-vault/lectures/14591190).

Complete the following Leetcode Problems and add them to your LC Review Schedule.

|  |
| --- |
| **Problems Name / Link** |
| 1. [Sort an Array](https://leetcode.com/problems/sort-an-array/) |
| 1. [Merge Intervals](https://leetcode.com/problems/merge-intervals/) |
| 1. [Merge Sorted Array](https://leetcode.com/problems/merge-sorted-array/) |
| 1. [Insertion Sort List](https://leetcode.com/problems/insertion-sort-list/) |
| 1. [Sort Colors](https://leetcode.com/problems/sort-colors/) |