CSE 332 Winter 2020 **Quiz 8** Name: UW NetID:

1 Comparison Sorts

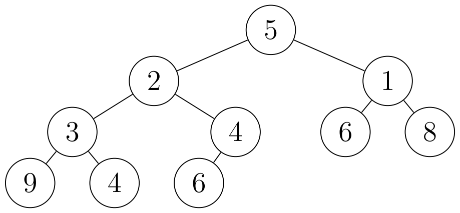
a) How many inversions are there in $3, 2, 5, 7, 6, 4, 2$? _____

b) Run the **insertion sort** on $3, 2, 5, 7, 6, 4, 2$. The first call of `swap` on this array is `swap(0, 1)` which means swap the elements in indices `0` and `1`. Fill in the next five `swap` operations to continue the insertion sort.

2nd Call: `swap(___,___)`
3rd Call: `swap(___,___)`
4th Call: `swap(___,___)`
5th Call: `swap(___,___)`
6th Call: `swap(___,___)`

- How many more `swap` operations do we need to call to finish the insertion sort? _____

c) Given an array and its representing tree below, perform **bottom-up max-heapification** to construct a max-heap. Show the heap and the array representation after the heapification.



Initial:	5	2	1	3	4	6	8	9	4	6
Result:										

d) Give **two** reasons why bottom-up heapification is better than constructing the heap naively.

e) Using the idea of merging from **mergesort**, can these two arrays be merged? If so, give the merged result. If not, briefly explain why and fix the original arrays.

3	5	7	6

1	3	4	4

f) Given an array A of size $2N$ with N items in sorted order in indices 0 through $N-1$, and an array B of size N with N items in ascending order, describe an algorithm in English to merge the array B into A so that A contains all of the items in ascending order. Only $O(1)$ **extra memory** allowed. Also, justify the **runtime** of your algorithm.

Will you want to pick up your worksheet later? Circle one: Yes / No
How confident do you feel with the material this week? Circle one: 1 / 2 / 3 / 4

g) Perform **Hoare partitioning** on 3,7,**5**,8,6,2,2 with 5 as a pivot. Fill in `swap` operations and intermediate results until we finish the partitioning. The first `swap` is provided for you to keep the pivot safe at the leftmost index.

1) `swap(0, 2)`	5	7	3	8	6	2	2
2) `swap(__,__)`	__	__	__	__	__	__	__
3) `swap(__,__)`	__	__	__	__	__	__	__
4) `swap(0,__)`	__	__	__	__	__	__	__

h) Give one advantage and one disadvantage of Hoare partitioning, compared to naive partitioning.

Advantage:
Disadvantage:

i) Assuming a weird partitioning runs in $\Theta(N^2)$ where N is the size of an array to be partitioned, give the recurrences representing the best-case runtime and the worst-case runtime of **quicksort** using this weird partitioning.

Best-case: $T(N) =$ ________________ Worst-case: $T(N) =$ ________________

2 Radix Sorts

a) Show the output after **three** passes in least significant digit (**LSD**) radix sort for the input $[5423, 452_{\text{A}}, 3156, 299, 956, 452_{\text{B}}]$, i.e., after applying counting sorts three times starting from the rightmost digit. Note that, A and B just tell the order of which one comes first.

______ ______ ______ ______ ______ ______

b) One TA wants to experiment with LSD radix sort and MSD radix sort by replacing counting sort with different sorting algorithms and run each radix sort as usual using the chosen sorting algorithm. Select sorting algorithm(s) to replace counting sort such that each radix sort will give correct result in any given input.

- LSD radix sort:
  ☐ Selection Sort ☐ Insertion Sort ☐ Heap Sort ☐ Mergesort
  ☐ Naive Quicksort ☐ Quicksort with Hoare Partitioning
- MSD radix sort:
  ☐ Selection Sort ☐ Insertion Sort ☐ Heap Sort ☐ Mergesort
  ☐ Naive Quicksort ☐ Quicksort with Hoare Partitioning

Will you want to pick up your worksheet later? Circle one: Yes / No
How confident do you feel with the material this week? Circle one: 1 / 2 / 3 / 4