CSE214 – Spring 2023
Recitation #12

1a. [10 minutes]

	Worst Case	Average Case	Best case
Binary search of a sorted array	$O(\log n)$	$O(\log n)$	$O(1)$
Insertion sort	$O(n^2)$	$O(n^2)$	$O(n)$
Merge Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Quick Sort without "median of three" pivot selection	$O(n^2)$	$O(n \log n)$	$O(n \log n)$
Bubble sort	$O(n^2)$	$O(n^2)$	$O(n)$
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Heap sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Counting sort	$O(n+k)$	$O(n+k)$	$O(n+k)$

1b. When does the best case for bubble sort and insertion sort occur?

Sorted

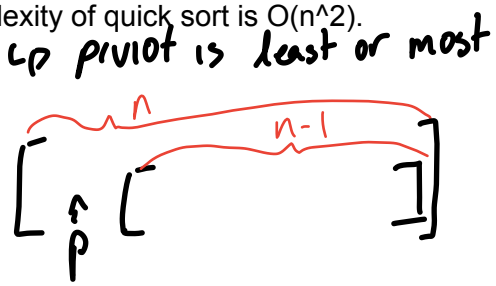
1c. Show that the best case for bubble sort and insertion sort is O(n).

1, 2, 3, 4, 5

↑ ↑ ↑ ↑

$O(n)$

2a) Show that the worst case time complexity of quick sort is O(n^2).



2b) Show that the time complexity of heap sort is O(n log n).

Build heap + For each element remove from heap

$O(n)$ $O(n \log n)$ = $O(n \log n)$

each remove.

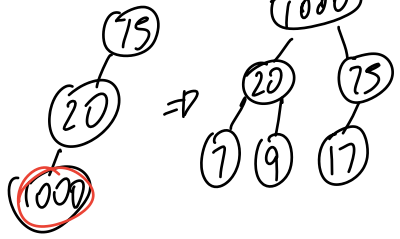
3) Perform Merge Sort on the following array: [66, 22, 32, 15, 28, 43, 29, 80]

[66, ...][28, ...]

[66][22][32][15][28][43][29][80]
[22,66] [15,32] [28,43][29,80]
[15,22,32,66][28,29,43,80]
[15,22,28,29,32,43,66,80]

4.[10 minutes] Perform Heap Sort to sort the array `[75, 20, 1000, 7, 9,17]` in ascending order.

1) Build heap.



=> [1000,20,75,7,9,17]

[75,20,17,7,9|1000]

[20,9,17,7|75,1000]

[17,9,7|20,75,1000]

[9,7|17,20,75,1000]

[7|9 17,20,75,1000]

[7,9,17,20,75,1000]

4a.
What are the number of operations to convert an array into a heap?

Build Heap

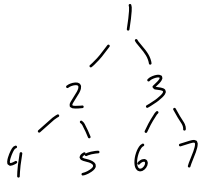
O(n)

4b.
For a heap array, what is the formula to find the parent, left child, right child?

current node : i
parent : (i-1)/2
left child : 2i+1
right child : 2i+2

4c.
Is an array sorted in ascending order always a min-heap? Why or why not?

[1,2,3,4,5,6]



Yes, parent is allways less than child.

4d. [75, 20, 1000, 7, 9, 17]

How would the first first iteration of quicksort look on this array if we have the pivot be the last element? [75, 20, 1000, 7, 9, | 17]

[7, 20, 1000, 75, 9 | 17]

[7, 9, 17, 15, 20, 1000]

. . . .

5. [5 minutes] Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array look like this:

2 5 1 7 30 12 11 10

Which statement is correct?

A. The pivot could be either the 7 or the 30
B. The pivot could be the 7, but it is not the 30
C. The pivot is not the 7, but it could be the 30
D. Neither the 7 nor the 30 is the pivot

[(less) p (greater)]

Thus, 30 cannot be pivot

7 can be