CS 61BL Summer 2021

Sorting

Quiz 12: Friday August 6, 2021

1 Best Sort

(a) Which sort is best for nearly sorted arrays?
( ) Selection sort ( ) Insertion sort ( ) Quicksort

What would be the runtime of your selected algorithm on the nearly sorted array? Let n be the number of elements in the array.

[ ] $\Theta(\log(n))$
[ ] $\Theta(n)$
[ ] $\Theta(n\log(n))$
[ ] $\Theta(n^2)$

(b) Each student in our course has a unique ID of length 10 which are allocated by alphabetizing the students by their names and counting off.

If we wanted to sort our student objects, what would be the fastest sort you could use to achieve this?

( ) Insertion sort ( ) Quicksort ( ) Radix sort

What would be the runtime of your selection above?

[ ] $\Theta(\log(n))$
[ ] $\Theta(n)$
[ ] $\Theta(n\log(n))$
[ ] $\Theta(n^2)$

2 Mechanical

For all parts, input each element in the array in their correct order with a **single space** between each element.

For example, if the array contained {`1, 2, 3`}, you would input `1 2 3`. For all parts below, we are sorting the same array. Also note that Quicksort and MSD Radix sort aren't covered below, but they are in scope for the final.

(a) Suppose we are sorting the array {`7, 38, 6, 94, 82, 19`}.

After **6** swaps of **in-place heapsort using a max heap**, what is the state of our array?

(b) Suppose we are sorting the array {`7, 38, 6, 94, 82, 19`}.

After **3** swaps of **selection sort**, what is the state of our array?

(c) Suppose we are sorting the array {`7, 38, 6, 94, 82, 19`}.

After **3** swaps of **insertion sort**, what is the state of our array?

(d) Suppose we are sorting the array {`7, 38, 6, 94, 82, 19`} with `mergeSort`.

Before the final merge pass, what is the state of the two arrays? Select the *two* arrays below. Assume if we divide an odd length array in half, the right half is longer.

[ ] (6 7 38)
[ ] (6 7)
[ ] (38 82 94)
[ ] (7 38)
[ ] (19 82 94)
[ ] (82 94)
[ ] (7 38 6)
[ ] (94 82 19)

(e) Suppose we are sorting the array {`7, 38, 6, 94, 82, 19`} with `LSD sort`.

After the *first* pass, what is the state of the array?