Sheet 4
Sorting

1 Question 1

Given an array of integers: 50, 28, 20, 26, 9, 78, 59, 94, do the following.

a) Sort the values (showing the procedure, step-by-step), using merge sort.

b) Run the quick sort, using median of three with first, last, and middle values. Show the step-by-step procedure.

c) Given that merge sort's worst-case time is better than quicksort's, why is quicksort so commonly used in practice?

2 Question 2

a) Demonstrate the actions of Shell Sort for the list of numbers given: 6, 56, 34, 23, 16, 4, 8, 1, 30, 41, 37, 52, 2

b) Sort the elements of the array [Cat, Bit, Hat, Mat, Rat, Bat, Hot, Bot, Sit] using top-down merge sort approach. Show all operations.

c) The Egyptian national flag problem is to rearrange an array of characters R, W, and B (red, white, and black are the colors of the Egyptian national flag) so that all the R's come first, the W's come next, and the B's come last. Explain which sorting algorithm is the best to solve the Egyptian national flag problem. Explain how.

3 Question 3

Assume we have a system of network connections and we're utilizing and data received from a connection using quick sort. We wish to protect our system from being "sabotaged" by hostile connections - we may get data that is purposefully designed to force quick sort to perform at its worst-case, causing our system to waste unnecessary resources and time (rendering it unable to efficiently respond to other connections).

a) Assuming that quick sort simply chooses the first element as the pivot, what is the arrangement of data that produces the worst-case performance in quick sort? (That is, if you were the attacker trying to sabotage the system, what data would you have to send?)

b) Suggest a simple strategy (hopefully requiring no more than linear time) to avoid the problem. That is, a strategy to guarantee that quick sort will run in O(n log n) most of the time, regardless of input data, even if this input data is maliciously created. Notice that in the context of this question, you're not allowed to change the way quick sort selects the pivot (in fact, you will hopefully suggest a strategy that works regardless of how quick sort selects the pivot).

c) A simpler technique for finding the pivot is picking an arbitrary value, such as the first one or the last one. Show for each of these two cases, a sequence of values that causes quick sort to achieve its worst-case run time (i.e., a sequence of values that makes it run in $O(n^2)$)

4 Notes

- You are required to submit a PDF of your answers and your ID in teams before 11:59 AM.
- You are encouraged to ask any questions on teams, or in person.

Good Luck