CS 61B Fall 2025

Sorting II

Discussion 12: November 17, 2025

1 Sort Identification

Match the sorting algorithms to the sequences, each of which represents several intermediate steps in the sorting of an array of integers. Assume that for quicksort, the pivot is always the first item in the sublist being sorted. Note that these steps are not necessarily the first few intermediate steps and there may be steps which are skipped.

Algorithms: *Quicksort, Merge Sort, Heapsort, Insertion Sort*

0.1
```
12, 7, 8, 4, 10, 2, 5, 34, 14
7, 8, 4, 10, 2, 5, 12, 34, 14
4, 2, 5, 7, 8, 10, 12, 14, 34
```

0.2
```
23, 45, 12, 4, 65, 34, 20, 43
4, 12, 23, 45, 65, 34, 20, 43
```

0.3
```
45, 23, 5, 65, 34, 3, 76, 25
23, 45, 5, 65, 3, 34, 25, 76
5, 23, 45, 65, 3, 25, 34, 76
```

0.4
```
23, 44, 12, 11, 54, 33, 1, 41
54, 44, 33, 41, 23, 12, 1, 11
44, 41, 33, 11, 23, 12, 1, 54
```

2 Quicksort

1.1 Sort the following unordered list using Quicksort. Assume that we always choose first element as the pivot and that we use the 3-way merge partitioning process described in lecture. Show the steps taken at each partitioning step.

```
18, 7, 22, 34, 99, 18, 11, 4
```

1.2 What is the best and worst case running time of Quicksort with Hoare partitioning on N elements? Given the two lists `[4, 4, 4, 4, 4]` and `[1, 2, 3, 4, 5]`, assuming we pick the first element as the pivot every time, which list would result in better runtime?

1.3 What are two techniques that can be used to reduce the probability of Quicksort taking the worst case running time?

3 Conceptual Sorts

Answer the following questions regarding various sorting algorithms that we've discussed in class. If the question is T/F and the statement is true, provide an explanation. If the statement is false, provide a counterexample.

2.1 (T/F) Quicksort has a worst case runtime of $\Theta(N \log N)$, where N is the number of elements in the list that we're sorting.

2.2 We have a system running insertion sort and we find that it's completing faster than expected. What could we conclude about the input to the sorting algorithm?

2.3 Give a 5 integer array that elicits the worst case runtime for insertion sort.

2.4 (T/F) Heapsort is stable.

2.5 Compare mergesort and quicksort in terms of (1) runtime, (2) stability, and (3) memory efficiency for sorting linked lists.

2.6 Describe how you might use a particular sorting algorithm to find the median of a list of N elements in worst case $\Theta(N \log N)$, without fully sorting the list.

2.7 You will be given an answer bank, each item of which may be used multiple times. You may not need to use every answer, and each statement may have more than one answer.

(A) Quicksort (in-place using Hoare partitioning and choose the leftmost item as the pivot)
(B) Merge Sort
(C) Selection Sort
(D) Insertion Sort
(E) Heapsort
(F) None of the above

For each of the statements below, list all letters that apply. Each option may be used multiple times or not at all. Note that all answers refer to the entire sorting process, not a single step of the sorting process, and assume that N indicates the number of elements being sorted.

_____ bounded by $\Omega(N \log N)$ lower bound.

_____ Worst case runtime that is asymptotically better than quicksort's worst case runtime.

_____ In the worst case, performs $\Theta(N)$ pairwise swaps of elements.

_____ Never compares the same two elements twice.

_____ Runs in best case $\Theta(\log N)$ time for certain inputs.