Analytical Questions on Sorting and Searching: PS-3

Suchintan Mishra

Department of Computer Science and Engineering
Institute of Technical Education and Research,
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India

Text Books

Text book and References:

- (T1) Algorithm Design by Jon Kleinberg and Eva Tardos, Pearson Publication Reference book
- (R1) **The Algorithm Design Manual by Steven Skiena, Springer Publication**
- (R2) Introduction to Algorithms by CLRS, PHI Publication

Outline

Merge Sort

Using MergeSort sort the sequence of characters:

M E R G E S O R T

Show the Steps involved in Divide and Conquer

Heap

Give an efficient algorithm to find the second-largest key among n keys. You can do better than $2n - 3$ comparisons.

Misc Sorting

Consider the problem of sorting a sequence of n 0's and 1's using comparisons. For each comparison of two values x and y, the algorithm learns which of $x < y$, $x = y$, or $x > y$ holds.

- Give an algorithm to sort in $n - 1$ comparisons in the worst case. Show that your algorithm is optimal.
- Give an algorithm to sort in $2n/3$ comparisons in the average case (assuming each of the n inputs is 0 or 1 with equal probability). Show that your algorithm is optimal.

Lower Bound

[5] In one of my research papers [Ski88], I discovered a comparison-based sorting algorithm that runs in $O(n \log(\sqrt{n}))$. Given the existence of an $\Omega(n \log n)$ lower bound for sorting, how can this be possible?

[Ski88] S. Skiena. Encroaching lists as a measure of presortedness. BIT, 28:775–784, 1988.

Application of Sorting I

Given a set S_1 a number x, describe an O(n log n) algorithm for finding whether there exists a pair of elements, from S_1 that add up to x. (For partial credit, give a (n^2) algorithm for this problem.)

Application of Sorting II

The mode of a set of numbers is the number that occurs most frequently in the set. The set (4, 6, 2, 4, 3, 1) has a mode of 4. Give an efficient and correct algorithm to compute the mode of a set of n numbers.

Application of Sorting III

You are given a set S of n intervals on a line, with the *ith* interval described by its left and right endpoints (l_i, r_i). Give an $O(nlogn)$ algorithm to identify a point p on the line that is in the largest number of intervals.

As an example, for $S = (10, 40), (20, 60), (50, 90), (15, 70)$ no point exists in all four intervals, but $p = 50$ is an example of a point in three intervals. You can assume an endpoint counts as being in its interval.

Searching I

A company database consists of 10,000 sorted names, 40% of whom are known as good customers and who together account for 60% of the accesses to the database. There are two data structure options to consider for representing the database:

- Put all the names in a single array and use binary search.
- Put the good customers in one array and the rest of them in a second array. Only if we do not find the query name on a binary search of the first array do we do a binary search of the second array.

Searching II

Suppose that you are given a sorted sequence of distinct integers $a_1, a_2, ..., a_n$. Give an O(lg n) algorithm to determine whether there exists an i index such as $a_i = i$. For example, in $-10, -3, 3, 5, 7$, $a_3 = 3$. In 2, 3, 4, 5, 6, 7, there is no such i.

Problem of Scale

If you are given a million integers to sort, what algorithm would you use to sort them? How much time and memory would that consume?

Unique elements

Implement an algorithm that takes an input array and returns only the unique elements in it.

Coin Weightage

You are given 12 coins. One of them is heavier or lighter than the rest. Identify this coin in just three weighings.

For Programming Freaks

Programming Challenges

These programming challenge problems with robot judging are available at *http://www.programming-challenges.com* or *http://online-judge.uva.es*.

4-1. "Vito's Family" – Programming Challenges 110401, UVA Judge 10041.

4-2. "Stacks of Flapjacks" – Programming Challenges 110402, UVA Judge 120.

4-3. "Bridge" – Programming Challenges 110403, UVA Judge 10037.

4-4. "ShoeMaker's Problem" – Programming Challenges 110405, UVA Judge 10026.

4-5. "ShellSort" – Programming Challenges 110407, UVA Judge 10152.

Thank You