HW

Jack Ruder

September 13, 2024

1 Problem 1

Answer

2 Problem 2

Claim: An array of integers $A[1...n]$ can be sorted in $O(n + D)$ time, where $D = \max_i A[i] - \min_i A[i]$ (allowing for use of extra storage).

Proof. First, see the following algorithm. The key observation is that there are at most $D + 1$ distinct values in A. This allows us to count the number of occurrences of each value, and then fill in the sorted array by iterating over the counts.

Algorithm 1 Sort an array of integers in $O(n + D)$ time.

Input: Array $A[1 \dots n]$, $D = \max_i A[i] - \min_i A[i]$, $A[i] \in \mathbb{Z}$
Output: Sorted array $A[1 \dots n]$

1: Find $a_{min} = \min_i A[i]$ and $a_{max} = \max_i A[i]$. $O(n)$
2: ▷ *Count the number of occurrences of each value in A*
3: Initialize $C[1 \dots D + 1]$ as an array of $D + 1$ zeros. $O(D + n)$
4: **for** $j \in \{1 \dots n\}$ **do**
5: $\quad i \leftarrow A[j] - a_{min} + 1$ ▷ *ex. $C[1]$ is the number of a_{min} seen in A*
6: $\quad C[i] \leftarrow C[i] + 1$
7: **end for** $O(n)$
8: ▷ *Fill in the sorted array*
9: Initialize $S[1..n]$ as en empty array. $O(n)$
10: $start \leftarrow 1$
11: **for** $i \in \{1, \dots D + 1\}$ **do**
12: $\quad end \leftarrow start + C[i]$ ▷ *find the contiguous blocks of S*
13: $\quad d \leftarrow a_{min} + i - 1$ ▷ *d is the value as seen in A*
14: $\quad$ FILL$(S[start \dots end], d)$ ▷ *populate S with d in the given range*
15: $\quad start \leftarrow end + 1$
16: **end for** $O(D + n)$
17: **return** S

We proceed to prove the runtime of the algorithm. The algorithm first finds the minimum and maximum values in the array, which trivially is $O(n)$ time. Next, the algorithm initializes an array of size $D + 1$ with zeros, which takes $O(D) \subset O(D + n)$ time. The algorithm then counts the number of occurrences of each value in the array using $O(n)$ time. To examine the loop on line 11, see that $C[i] = end_i - start_i$ where end_i and $start_i$ are the values of the variables at line 12 of the loop. We have $C[i]$ as the size of the contiguous block of S. Observe that $\sum_i C[i] = n$. Then, since each call to FILL takes $O(end - start)$ time, the total running time of the work FILL performs is

$$\sum_i^{D+1} O(end_i - start_i) = \sum_i^{D+1} O(C[i]) = O(n).$$

Since there are $D + 1$ iterations of the loop, the total running time of the loop outside of the FILL function is $O(D)$. Therefore, the running time of the loop is $O(n + D)$, allowing us to conclude that the running time of the entire algorithm is $O(D + n)$.

□

3 Problem 3

You are interested in analyzing some remote data from two separate databases. Each database contains n numerical values—so there are 2n values total, and you may assume that no two values are the same. You'd like to determine the median of this set of $2n$ values, which we will define here to be the nth smallest value. However, the only way you can access these values is through queries to the databases. In a single query, you can specify a value k to one of the two databases, and the chosen database will return the kth smallest value that it contains. Since queries are expensive, you would like to compute the median using as few queries as possible.

1. Devise an algorithm that finds the median value using at most $O(\log n)$ queries. Explain why your algorithm achieves this query bound.
2. Write a small program to test the method that you devised. You may assume that the databases are just two sorted arrays and you want to make $O(\log n)$ accesses to those arrays. Demonstrate it running on an example.

(1) Answer: [[**TODO:** Probably need to do the following:

- Describe the algorithm
- Prove correctness/termination
- Analyze the running time

]]

(2) Answer: [[**TODO:** Include code demonstration/figures]]

4 Problem 1

Answer

5 Problem 1

Answer

6 Problem 1

Answer