

Rank Transform Of An Array

🕒 solved by	Senan
🌐 Platform	LeetCode
🔧 difficulty	Easy
# Serial	1331
≡ tags	Hash Map Sorting Vector
🗨 language	C++
📅 solved on	@October 2, 2024
🔗 link	https://leetcode.com/problems/rank-transform-of-an-array/
☑ Completion	✓

Intuition

The goal is to rank the elements of an array such that the smallest element gets the rank 1, the second smallest gets rank 2, and so on. Duplicates should receive the same rank. The key idea is to map the unique elements to their respective ranks, then replace each element in the original array with its rank.

Approach

- Use a set to sort and remove duplicates:** By inserting all elements of the array into a set, we can automatically remove duplicates and get the elements in sorted order.
- Assign ranks to each unique element:** Iterate through the sorted set and assign a rank (starting from 1) to each element, storing these in a hash map (unordered_map). The keys are the elements, and the values are their corresponding ranks.
- Replace elements with their ranks:** Finally, traverse the original array and replace each element with its corresponding rank from the map.

Complexity

Time Complexity:

- **$O(N \log N)$:**
 - Creating the set takes $O(N \log N)$ time because inserting N elements into a set involves sorting them.
 - Iterating over the set and original array takes $O(N)$ time.
- Thus, the total time complexity is dominated by $O(N \log N)$.

Space Complexity:

- **$O(N)$:**
 - The extra space is used for the set and the hash map, both of which store at most N elements.

Code

```
class Solution {
public:
    vector<int> arrayRankTransform(vector<int>& arr) {
        // Use set to sort and remove duplicates
        set<int> mySet(arr.begin(), arr.end());

        int rank = 0;
        unordered_map<int,int> mpp;

        // Assign ranks to the sorted elements
        for(auto elem: mySet) mpp[elem] = ++rank;

        // Replace array elements with their corresponding rank
        for(int i = 0; i < arr.size(); i++) arr[i] = mpp[arr[i]];

        return arr;
    }
};
```