Alternate Sorting

solved by	Senan
Platform	GeeksForGeeks
difficulty	Easy
tags	Array Sorting Vector
language	C++
solved on	@25/10/2024
link	https://www.geeksforgeeks.org/problems/alternative-sorting1311/1
Completion	☑

Intuition

The goal is to alternate between the largest and smallest elements of the array, so the array will be sorted first. Then, starting from the largest and smallest, elements are added to a result array in pairs: one from the back (largest) and one from the front (smallest). If the array has an odd length, the middle element will be appended at the end.

Approach

1. Sort the array in non-decreasing order.
2. Initialize an empty result vector.
3. Loop through half of the array, and in each iteration:
    - Add the largest remaining element from the end of the sorted array.
    - Add the smallest remaining element from the front of the sorted array.
4. If the array has an odd number of elements, append the middle element to the result.
5. Return the result array.

Complexity

Time Complexity:

- Sorting the array takes **O(n log n)**, where **n** is the size of the array.
- Constructing the result array takes **O(n)** since we loop through half of the array and append elements.

Overall time complexity: **O(n log n)**.

Space Complexity:

- The space complexity is **O(n)**, where **n** is the size of the result array.
- No extra space is used apart from the result vector and the input array.

Code

```
class Solution {

```

```
  public:
    vector<int> alternateSort(vector<int>& arr) {
        vector<int> answer;
        sort(arr.begin(), arr.end());
        int n = arr.size();

        for(int i = 0 ; i < n / 2 ; i++){
            answer.push_back(arr[n - i - 1]);
            answer.push_back(arr[i]);
        }

        if(n & 1) answer.push_back(arr[n / 2]);

        return answer;
    }
};
```