Swap And Maximize

solved by	Senan
Platform	GeeksForGeeks
difficulty	Easy
tags	Sorting
language	C++
solved on	@01/11/2024
link	https://www.geeksforgeeks.org/problems/swap-and-maximize5859/1
Completion	☑

Intuition

The problem involves maximizing the sum of absolute differences between adjacent elements when arranging the elements of an array in a circular manner. The idea is to position the smallest and largest elements alternately to maximize the differences between adjacent elements.

Approach

1. **Sort the Array**: Start by sorting the array in ascending order.
2. **Rearrange Elements**: Create a new array `temp` by placing elements alternately from the start and end of the sorted array. This ensures that the difference between adjacent elements is maximized.
3. **Sum of Absolute Differences**: Iterate through the rearranged array `temp` and calculate the sum of absolute differences between consecutive elements, treating the array as circular by wrapping the last element to the first.
4. **Return the Result**: Return the calculated sum.

Complexity

Time Complexity:

- Sorting the array takes **O(n log n)**.
- Rearranging the elements and calculating the sum take **O(n)**.
- Overall, the time complexity is **O(n log n)**, where **n** is the length of the input array.

Space Complexity:

- The space complexity is **O(n)** due to the use of the `temp` array for rearranging elements.

Code

```
class Solution {
  public:
    long long maxSum(vector<int>& arr) {
        long long answer = 0;
        sort(arr.begin(), arr.end());
```

```
        int n = arr.size();
        vector<int> temp;

        int i, j;
        for(i = 0, j = n - 1; i < j; i++, j--) {
            temp.push_back(arr[i]);
            temp.push_back(arr[j]);
        }
        if(i == j) temp.push_back(arr[i]);

        for(int i = 0; i < n; i++) {
            answer += abs(temp[i] - temp[(i + 1) % n]);
        }

        return answer;
    }
};
```