Maximize sum(arr[i]*i) of an Array

Given an array **A** of **N** integers. Your task is to write a program to find the maximum value of *∑*arr[i]i*, where i = 0, 1, 2,…., n – 1.
You are allowed to rearrange the elements of the array.

Note: Since output could be large, hence module 10^9+7 and then print answer.

Example 1:

```
Input : Arr[] = {5, 3, 2, 4, 1}
Output : 40
Explanation: If we arrange the array as 1 2 3 4 5 then
we can see that the minimum index will multiply
with minimum number and maximum index will
multiply with maximum number.
So 1*0+2*1+3*2+4*3+5*4=0+2+6+12+20 = 40 mod(10<sup>9</sup>+7) = 40
```

Example 2:

```
Input : Arr[] = {1, 2, 3}
Output : 8
```

Your Task:
This is a function problem. The input is already taken care of by the driver code. You only need to complete the function **Maximize()** that takes an *array **(arr),** **sizeOfArray **(n)**, and return the maximum value of an array. The **driver code** takes care of the **printing**.

Expected Time Complexity: O(nlog(n)).
Expected Auxiliary Space: O(1).

Constraints:

$1 \le N \le 10^7$
$1 \le A_i \le N$

```python
class Solution:
    def Maximize(self, a, n):
        # Complete the function
        a.sort()
        sum_ = 0
        for i in range(n):
            sum_+=a[i]*i
        return (sum_)%(10**9+7)
```