Slow Sums

Suppose we have a list of **N** numbers, and repeat the following operation until we're left with only a single number: Choose any two numbers and replace them with their sum. Moreover, we associate a penalty with each operation equal to the value of the new number, and call the penalty for the entire list as the sum of the penalties of each operation.

For example, given the list **[1, 2, 3, 4, 5]**, we could choose **2 and 3** for the first operation, which would transform the list into **[1, 5, 4, 5]** and incur a penalty of **5**. The goal in this problem is to find the **highest** possible penalty for a given input.

Signature:

```
int getTotalTime(int[] arr)
```

Input:
An array **arr** containing **N** integers, denoting the numbers in the list.

Output format:
An int representing the highest possible total penalty.

Constraints:
1 ≤ **N** ≤ 10^6
1 ≤ **Ai** ≤ 10^7, where *_Ai_ denotes the **i**th initial element of an array.
The sum of values of **N** over all test cases will not exceed 5 * 10^6.

Example

```
arr = [4, 2, 1, 3]
output = 26
```

First, add **4 + 3** for a penalty of **7**. Now the array is `[7, 2, 1]`
Add **7 + 2** for a penalty of **9**. Now the array is `[9, 1]`
Add **9 + 1** for a penalty of **10**. The penalties sum to **26**.