6017. Append K Integers With Minimal Sum

My Submissions (/contest/weekly-contest-283/problems/append-k-integers-with-minimal-sum/submissions/)

Back to Contest (/contest/weekly-contest-283/)

You are given an integer array `nums` and an integer `k`. Append `k` **unique positive** integers that do **not** appear in `nums` to `nums` such that the resulting total sum is **minimum**.

Return *the sum of the* `k` *integers appended to* `nums`.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

```
Input: nums = [1,4,25,10,25], k = 2
Output: 5
Explanation: The two unique positive integers that do not appear in nums which we append are 2 and 3.
The resulting sum of nums is 1 + 4 + 25 + 10 + 25 + 2 + 3 = 70, which is the minimum.
The sum of the two integers appended is 2 + 3 = 5, so we return 5.
```

Example 2:

```
Input: nums = [5,6], k = 6
Output: 25
Explanation: The six unique positive integers that do not appear in nums which we append are 1, 2, 3, 4, 7, and 8.
The resulting sum of nums is 5 + 6 + 1 + 2 + 3 + 4 + 7 + 8 = 36, which is the minimum.
The sum of the six integers appended is 1 + 2 + 3 + 4 + 7 + 8 = 25, so we return 25.
```

Constraints:

- $1 <= nums.length <= 10^5$
- $1 <= nums[i], k <= 10^9$

Java

```java
class Solution {
    long sumOfRange (long l, long r) {
        return (l + r) * (r - l + 1) / 2;
    }

    public long minimalKSum(int[] a, int k) {
        TreeSet<Integer> ts = new TreeSet<>();
        for (int x: a) ts.add(x);
        long res = 0, l = 0, tot = 0;
        for (int r: ts) {
            long cnt = r - l - 1;
            if (cnt > 0) {
                long sum, start, end;
                if (tot + cnt >= k) {
                    long need = k - tot;
                    start = l + 1;
                    end = l + need;
                    sum = sumOfRange(start, end);
                    res += sum;
                    tot += cnt;
                    break;
                } else {
                    start = l + 1;
                    end = r - 1;
                    sum = sumOfRange(start, end);
                    res += sum;
                    tot += cnt;
                }
            }
```

```
            l = r;
        }
        if (tot < k) {
            long need = k - tot;
            res += sumOfRange(l + 1, l + need);
        }
        return res;
    }
}
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/654228857/) | More Details > (/submissions/detail/654228857/)

Share your acceptance!