Game with String

Given a string **s** of lowercase alphabets and a number k, the task is to print the minimum value of the string after removal of **‘k’** characters. The value of a string is defined as the sum of squares of the count of each distinct characte

Example 1:

Input: s = abccc, k = 1
Output: 6
Explaination: We remove c to get the value as 12 + 12 + 22

Example 2:

Input: s = aabcbcbcabcc, k = 3
Output: 27
Explaination: We remove two 'c' and one 'b'.
Now we get the value as 32 + 32 + 32.

Your Task:
You do not need to read input or print anything. Your task is to complete the function **minValue()** which takes s and k as input parameters and returns the minimum possible required value.

Expected Time Complexity: O(N*logN) where N is the length of string
Expected Auxiliary Space: O(N)

Constraints:
1 ≤ k ≤ |string length| ≤ 100

```
import collections
import heapq
class Solution:
    def minValue(self, s, k):
        # code here
        count = collections.Counter(s)
        count = sorted(count.items(),key=lambda x:-x[1])
        heap = []
        for i in range(len(count)):
            heap.append(-count[i][1])
        heapq.heapify(heap)
        while k>0:
            temp = heapq.heappop(heap)
```

```
            temp = temp+1
            heapq.heappush(heap,temp)
            k=k-1
        ans = 0
        while heap:
            ans = ans+pow(heapq.heappop(heap),2)
        return ans
```