Question: https://leetcode.com/problems/minimum-deletions-to-make-character-frequencies-unique/

so the idea is not something too complicated!

We store all the frequency is decreasing order ina queue, now what we do is we take one elemennt out of the queue and check if its next frequency is same as that or not, if same then we decrease the removed frequency and increase the counter annd re add the decreased frequency; and if not same then we simply continue without decreasing or re adding the frequency.

Now, repeat this until we encounter a 0 frequency or the queue is empty.

Code:

class Solution {

Map<Character, Integer> mem = new HashMap<>();

public int minDeletions(String s) {

int maxlen=s.length(), count=0;

for(int i=0; i<maxlen; i++){

char c = s.charAt(i);

mem.put(c, mem.getOrDefault(c, 0)+1);

}

PriorityQueue<Integer> queue = new PriorityQueue<>(Collections.reverseOrder()); ;

for (Map.Entry<Character,Integer> entry : mem.entrySet())

{

queue.add(entry.getValue());

}

while (queue.size()!=0) {

int most\_frequent = queue.poll();

if(queue.size()==0) break;

if(most\_frequent == 0) break;

if (most\_frequent == queue.peek()) {

most\_frequent--;

queue.add(most\_frequent);

count++;

}

}

return count;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>