[**https://www.hackerrank.com/challenges/sherlock-and-valid-string/problem?h\_r=next-challenge&h\_v=zen**](https://www.hackerrank.com/challenges/sherlock-and-valid-string/problem?h_r=next-challenge&h_v=zen)

**static** String isValid(String s) {

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

**int** count = s.length();

**int** Num0fChar = 0;

**int** freq = 0;

**for** (**char** c : s.toCharArray()){

**if** (!Map.containsKey(c)){

Map.put(c , 1);

Num0fChar++;

}**else** Map.put(c , Map.get(c) +1 );

}

freq = count % Num0fChar;

*// TreeMap to store values of HashMap*

TreeMap<Character, Integer> sorted = **new** TreeMap<>();

*// Copy all data from hashMap into TreeMap*

sorted.putAll(Map);

**int** min = Collections.min(sorted.values());

**int** max = Collections.max(sorted.values());

**if**((Math.abs(freq - Num0fChar) <= 2) &&((max-min)<2)){

**return** "YES";

}**else**{

**return** "NO";

}

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

**for(int i=0; i<s.length(); i++){**

**if(hm.containsKey(s.charAt(i))){**

**hm.put(s.charAt(i), hm.get(s.charAt(i))+1);**

**}else{**

**hm.put(s.charAt(i), 1);**

**}**

**}**

**int min = Collections.min(hm.values());**

**int max = Collections.max(hm.values());**

**if(min==max)**

**return "YES";**

**HashMap<Integer,Integer> intMap = new HashMap();**

**for(int count : hm.values())**

**{**

**Integer val = intMap.get(count);**

**if(null != val)**

**{**

**intMap.put(count,val+1);**

**}**

**else**

**{**

**intMap.put(count,1);**

**}**

**}**

**return intMap.size() == 2 && (intMap.get(min) == 1 || intMap.get(max) == 1) && (max-min<=1 || (min==1 && intMap.get(min) == 1))? "YES" : "NO";**

**}**