

	3) P	Yow	er	late	a by	ugu	eney	ar	rau	•	
	w	here	. ใข	rdex	nb	is	egha	lt	0	element	-
		nw	nlier	.							
	ь	١	2	3	4	5	6	7	8		
M=	0	1	1+1=2	11+1	1	1	0	0	1		
				-			•				

Replace the original away.
D 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
7 2 2 0 0 0 0 0
Dayan of +
Array Sorted
D'ime Complexity ' 0 (N+M) Opace complexity: 0 (N+M)
donce complexité · O (N+M)
Signal Solid Control
200 . I this u 1 1 1 0 c
We can also dothing thanh Mage?
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
4 - 1 =
1 - 1 2=0 tiU 8
2 - 2
$5 - 1 \operatorname{map. act}(9)$
5 -1 map. get (i)
1 2 2 3 3 4 5 6
Jime complexity: OCN+M)
Frace complexity: O(M)

Λ

public class count sout & // this is a stable sout Algorithm public static void count sout (int [Javer) { i) (averay == null averay length <= 1) { yetwin; }
<u>Algorithm</u>
public static void countBout (int [Javer) }
i) (array = = mill array length <= 1) $\sqrt{2}$
g yetun.
4
9 1
int max = soway [0]; // find the max element in the avoing
element in the evolay
ton (int nums: averay)
i) (nums > max) of // a) the coverent
8 max - nums: element is que ten
for (int nums: away) { i) (nums > max) f// if the current max = nums; element is quester than max
Than max
Int [] count Alonay = new int
[max +) Jj
// create a court average of size max+
ent [] count Averay = new int [max +)]; // create a count averay of size max+
Post part mine a mineral de
La Francis T
count showing Limited I ++;
// Court the frequency of each element
for (Int num: away) of count Amay [mun]] ++; // Count the treguency of each element in the advay
\mathcal{G}

Int index = 0; for (inti=0; z < = max; z + +) while count Amay [i] 70) {

while count Amay [i] 70) {

averay [Index] = 9.//

Replace the element in the amay with

the sorted element

court Amay [i]--;

3 I/we can also create count sout using flash Map to stone the frequency of each element in the away public static void count sout Hash Map (ent [] ceden) E If (away == Mull 11 away. length <=) \(\)
y Int max = Avorays. stream(avr).
max() getAsInt(); Int min = Asways. stream (avo).
min (). get As Int ().

11 now MapOut the beginning of each element In the lowery Map < Integer, Integer > count Map = new HashMap () 3 for (int num: anu) &

count Map. put (num, count Map.

get On Default (num, 0) + 1);//

If the number is not present, then add

default 0 or else increments the value

by 1

3 ent index = 0; for (int i = min; i <= max; i+) {

ent count = count Map. get On Default

(i, 0); //get the

frequency of the element. for (snt g = 0; g < count; j ++) [//
loop will hun that many times the number appeared
in the away if 2 appeared 2 times it will
run the loop for 2, two times

avoir [Index] = 2; //Replace Index ++; static void main (string[Jargs) int [] au = £4,2,2,8,3,13, // countsout (armay), count Sout Hasholdap (armay), System out print ln (Avrays to string (avray);

