

Hash Map & Heap Continued

Date			
Page No.			

14 March

- Grfg
1. Count of Pairs whose sum is divisible by K.

15	4	33	11	74	82	47	91	16	37	28	83
0	1	2	3	4	5	6	7	8	9	10	11

Total Pairs $\rightarrow 12C_2$ ie $nC_2 = \frac{n}{(n-2)12}$.

$$nC_2 = \frac{Ln}{Lm \cdot Lr}$$

$$nC_2 = \frac{12 \times 11 \times 10}{10 \times 9 \times 1}$$

$$= \frac{12 \times 11}{2} = 66$$

Approach 1 $\rightarrow O(n^2)$. Time Complexity.

nested for loop लगाओ, एक element की pair बनाओ
 दूसरे element की ओर check करें यदि sum is divisible by K or not & keep on
 increasing the count if it is divisible

Approach 2. $\rightarrow O(n)$ Time Complexity
 $O(n)$ Space Complexity

Eig 1

K=7

15 4 33 11 74 82 14 47 91 16 37 28 83.

Rem 1 4 5 4 4 5 0 5 0 2 2 0 6
n-K

We will make a hashmap to store the remainder with its frequency.

if $K=7$ then map	0 → 1 2 3	$\{ 7K+0 \}$	${}^3 C_2 = 3$
0 → 1	1 → 1	$\{ 7K+1 \}$	
0 → 2	2 → 1 2	$\{ 7K+2 \}$	
0 → 3	3 → 0	$\{ 7K+3 \}$	${}^3 P_3 = 6$
0 → 4	4 → 1 2 3	$\{ 7K+4 \}$	
0 → 5	5 → 1 2 3	$\{ 7K+5 \}$	
0 → 6	6 → 1	$\{ 7K+6 \}$	

$$= (3C_2) + (1 \times 1) + (2 \times 3) + (3 \times 0)$$

$$= \frac{3 \times 2}{2!} + 1 + 6 + 0$$

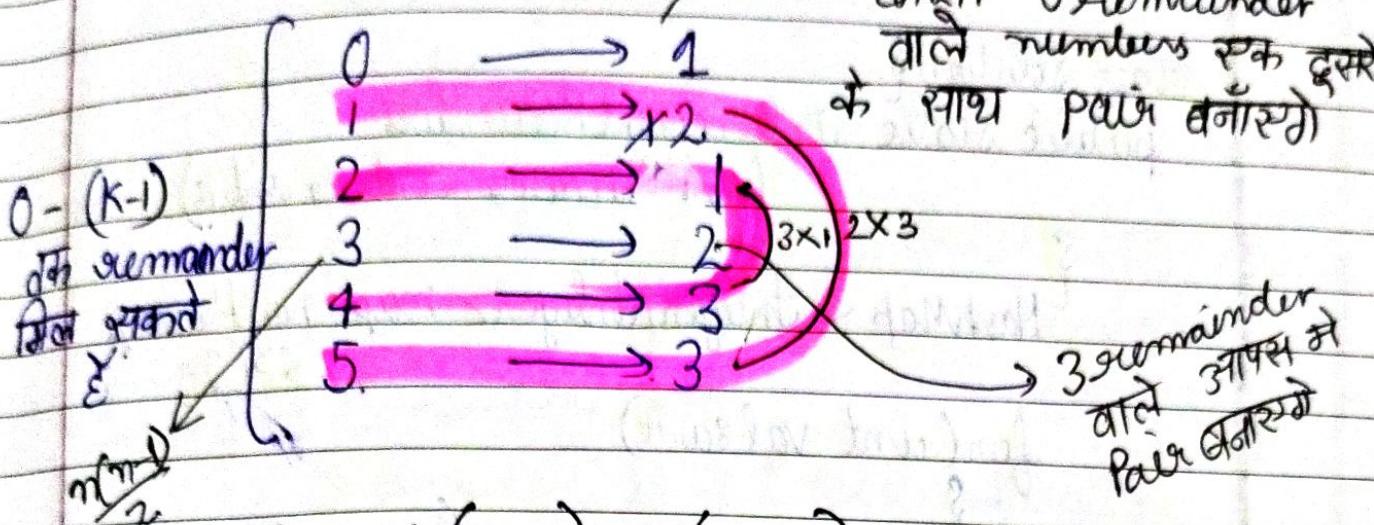
$$= 3 + 1 + 6 + 0$$

$$= 4 + 6$$
$$= 10.$$

Ex 2). $k=6$

15	4	33	11	74	82	47	91	16	37	28	83
Rem 3	4	3	5	2	0	5	1	4	1	4	5

Date	Page No.
------	----------



$$= 1C_2 + (3 \times 1) + (2 \times 3) + 1C_2$$

$$= 1 + (3 \times 1) + 6 + 1$$

$$= 1 + 3 + 6 + 1$$

$$= 10 + 1$$

2
fixed

Ex 3) If numbers are negative

$$K=5$$

-17	12	11	25	35	-21
↓	↓	↓	↓	↓	↓
-15 - 2X	2	1	0	0	-20 - 1X

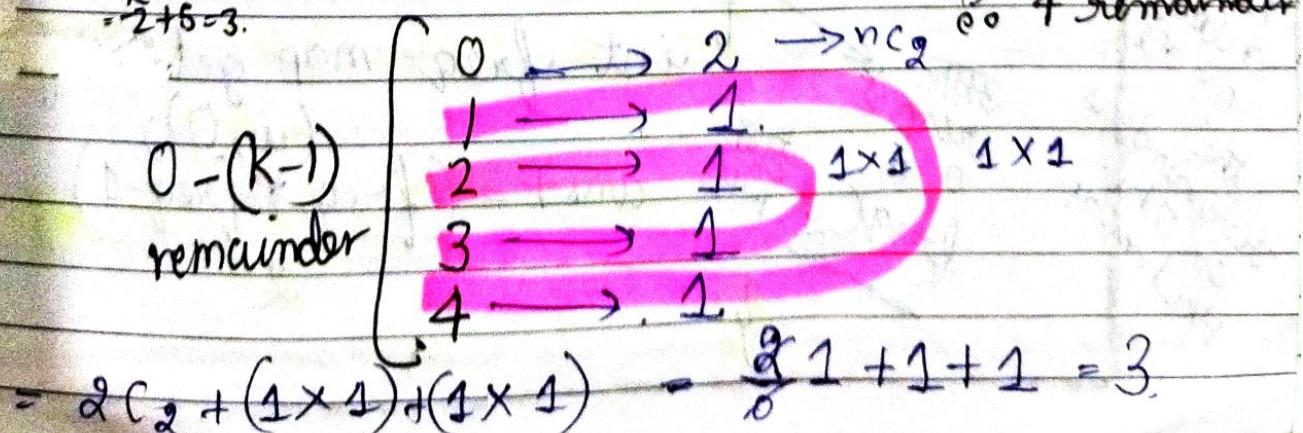
$$= -20 + 3. \checkmark$$

as sum = -2, then add

$$-2 + 5 = 3.$$

$$-25 + 4. \checkmark$$

$-1 + 5 - 4$
so 4 remainder



$$= 2C_2 + (1 \times 1) + (1 \times 1) - \frac{8}{5} 1 + 1 + 1 = 3$$

<code>

class Solution {

public static int countKdivPairs

(int[] arr, int n, int k).

{

 HashMap<Integer, Integer> map = new HashMap
<>();

 for (int val : arr)

 {

 int remainder = ((val % k) + k) % k;

We
will
store these
remainders with their
frequency in
HashMap.

It works for
all integers too

 int freq = map.getOrDefault

(remainder, 0);

 map.put(remainder, freq + 1);

 int ans = 0;

 for (int i = 0; i <= K / 2; i++)

 {

 if (i == 0)

 {

 int freq = map.getOrDefault

(i, 0);

 ans += (freq * freq - 1) / 2;

We
have
a loop
on all
the
remainders
in the
HashMap of

3 Pairs

rem

0

pair

freq

2

3rd remainder
if $\frac{K}{2}$ तो 3rd remainder
of other digits
का राहे एवं pair करें

Date _____
Page No. _____

else if ($i * 2 == K$)
int freq = map.getOrDefault($i, 0$);
ans += (freq * (freq - 1)) / 2;
freq[2] ;
else {
int freq = map.getOrDefault($i, 0$);
int secondfreq = map.getOrDefault($(K-i), 0$);
for rest of the numbers with
ans += (freq * secondfreq);
some other
remainder
will
also
make
pair
return ans;
}
this
will
be the
remainder of
other numbers

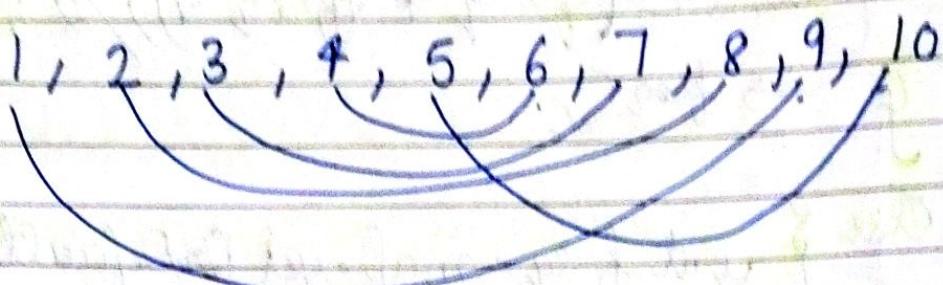
2. Check if array pairs are divisible by K.

Eg:-

उमे क्ताना
है कि या

इस array
में $\frac{n}{2}$ pairs

जिनके यही
sum are divisible
by K.



All these pairs will be
divisible by 5

If K=5, remainders can be 0, 1, 2, 3, 4.

remainder

1	1
2	2
3	3
4	4
5	0
6	1
7	2
8	3
9	4
10.	0

0 remainder
of total numbers
आपस में pair
जिनके only
if they are
even,
otherwise return false

if remainder वाले numbers
K-वाले remainders वाले numbers
के साथ pair जाना सकते हैं

only if both are equal in
numbers, otherwise
just return false.

K remainder वाले numbers
 $\frac{2}{2}$ दोनों K remainder

वाले numbers के साथी
pair जाना सकते हैं, only if
frequency of

$\frac{K}{2}$ remainders are even in number

~~Ex 2~~

$K = 10$

12	21	34	23	45	78	39	56	47	55
\downarrow									

$\text{rem} = 2, \text{rem} = 1, \text{rem} = 4, \text{rem} = 3, \text{rem} = 5, \text{rem} = 8, \text{rem} = 9, \text{rem} = 6, \text{rem} = 7, \text{rem} = 5$

rem	frequency	freq of rem x is same as $K-x$ so return true because there are
2	1	
1	1	
4	1	
3	1	
5	1	
8	1	
9	1	
6	1	
7	1	

freq of 0 remainder is 0. $\frac{n}{2}$ pairs.

Approach

- ① Array of ~~all~~ values \rightarrow remainders calculate ~~freq~~ !
- ② Put the remainder with its frequency in Hashmap.
- ③ Now we will put a loop from 0 to $\leq K/2$ remainder of Hashmap.

If all these conditions satisfies it means there are $\frac{n}{2}$ pairs -

\rightarrow if freq of $\text{rem} = 0$ is even \rightarrow continue

\rightarrow if freq of $\text{rem} = K$ is even \rightarrow continue

\rightarrow if freq of x remainder == freq of $K-x$ remainder \rightarrow continue

Otherwise return false.

If remainder is -ve.

eg \rightarrow number = -17
 $K = 5.$

remainder = -2.

So how will we calculate remainder for -ve.

$$= ((\text{num} \% K + K) \% K)$$

$$= (-2 + 5) \% 5$$

$$= (3) \% 5$$

$$= 3$$

<code>

Page No.

class Solution

{ public static int canArrange(int arr, int K)

{ HashMap< Integer, Integer > map = new HashMap<>();

for (int val : arr)

{

int remainder = (val % K) + K) % K;

int freq = map.getOrDefault(remainder, 0);

map.put(remainder, freq + 1);

}

for (int val : arr).

{

if (i == 0).

{

int freq = map.getOrDefault(0, 0);

if (freq % 2 != 0)

{

return false;

}

frequency
of 0 remainder
should
be even.

else if (i == K) { int freq = map.getOrDefault(

($\frac{K}{2}$, 0));

if (freq % 2 != 0)

{ return false; }.

frequency
of $\frac{K}{2}$
remainder
should
be even.

for rest of
the remainders

Date _____
Page No. _____

```
else { int freq = map.getOrDefault(i, 0);  
int secondfreq = map.getOrDefault(k-i, 0);
```

```
if (freq != secondfreq)  
    return false;
```

```
}  
return true;
```

Leetcode 914

Date			
Page No.			

ques. 3

X of a Kind in a deck of Cards

→ ऐसे एक array दिया जाएगा.

→ we need to return true iff.

→ हमें array को groups में divide करना है ताकि
such that a) एक group में x card हों



$$x \geq 2$$

b) 1 group में सभी cards same
हों।

eg → [1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3] → 8 1's.
→ 12 2's
→ 16 3's.

1) इन values को hashmap में number store करेंगे उनके frequencies.

$$1 \rightarrow 8$$

$$2 \rightarrow 12$$

$$3 \rightarrow 16$$

3. map::values().set द्वारा दिये गए values(frequency)

प्रत्येक value के लिए 8, 12, 16

find greatest common divisor
find gcd of all of them

$$\text{gcd} = 4$$

8 can be divided into 2 set of 4. [1, 1, 1, 1] [1, 1, 1, 1]

12 can be divided into 3 sets of 4 [2, 2, 2, 2] [2, 2, 2, 2] [2, 2, 2, 2]

16 can be divided into 4 sets [3, 3, 3, 3] [3, 3, 3, 3] [3, 3, 3, 3], [3, 3, 3, 3]

Approach have to

So we will find true or false

1 set should have same value

Each set should have the size ≥ 2 & same size

Approach

① So, first of all, we will make a frequency map.

$$\begin{array}{l} a \rightarrow 24 \\ b \rightarrow 36 \\ c \rightarrow 48 \\ d \rightarrow 60 \end{array} \quad \begin{array}{l} \xrightarrow{12} \\ \xrightarrow{12} \\ \xrightarrow{12} \end{array}$$

$$\begin{array}{l} [aaaaaaa] \times 2 \\ [bbbbbbbbb] \times 3 \\ [ccccccc] \times 4 \\ [ddddd] \times 5 \end{array}$$

$\text{gcd} = \text{greatest common divisor}$

If we are able to find a
gcd of all frequencies ≥ 2 .

Then, we can return true,
else return false.

< code >

```

class Solution {
    public boolean hasGroupsSizeX(int[] deck) {
        {
            HashMap<Integer, Integer> map = new HashMap<>();
            for (int i = 0; i < deck.length; i++) {
                int freq = map.getOrDefault(deck[i], 0);
                map.put(deck[i], freq + 1);
            }
            int counter = 0;
            int gcd = 1;
            for (int key : map.keySet()) {
                if (counter == 0) {
                    gcd = map.get(key);
                } else {
                    gcd = getGCD(gcd, map.get(key));
                }
                counter++;
            }
            return gcd >= 2;
        }
    }

    private int getGCD(int gcd, int freq) {
        if (freq % gcd == 0) {
            return gcd;
        } else {
            return getGCD(freq, gcd);
        }
    }
}

```

To keep track on key no. in the set

We are putting a loop on each key

so that we can find gcd of their frequencies

return gcd ≥ 2

public static int getGCD (int first, int second)

This
is the
function
to
calculate
GCD.

{
while (first % second != 0)

{
int x = first % second;
first = second;
second = x;

}
return second;

Date _____
Page No. _____

Ques. 4 Check AP (Arithmetic Progression)

→ To check if there is an AP sequence.

AP \Rightarrow AP sequence follows this formula

$$a_n = a_1 + (n-1)d$$

↓ ↓ ↓
 nth digit first (minimum) digit distance

Check AP \rightarrow 2 7 12 17 22 27 32

Is it AP?

Approach

- ① Find the minimum & second minimum in the array & push all elements of array in a Hashmap / HashSet
→ first digit of AP sequence

② $a_1 = \min$

$d = (\text{second min}) - \min$

→ then check whether the length of array is same as AP sequence.

→ Now create an array of same length.

Now

e.g. given array to check AP sequence

2 7 17 22 12 27 37

Check whether it is AP.

(a) Now put all elements of array in a HashSet

↳ 2
7
17
22
12
27
37

HashSet

↳ HashSet जिसमें keys store होते हैं ताकि values नहीं।

(b) also find min & second minimum in array.

↳ first element of AP

(c) Now, $a = \text{minimum}$
 $d = \text{second minimum} - \text{minimum}$
 ↳ difference in AP sequence.

2 7 17 22 12 27 37

↑

minimum

↑

second

minimum

$$a = 2$$

$$d = 7 - 2$$

→ We have a , d and we also have given array



check the length
of given array

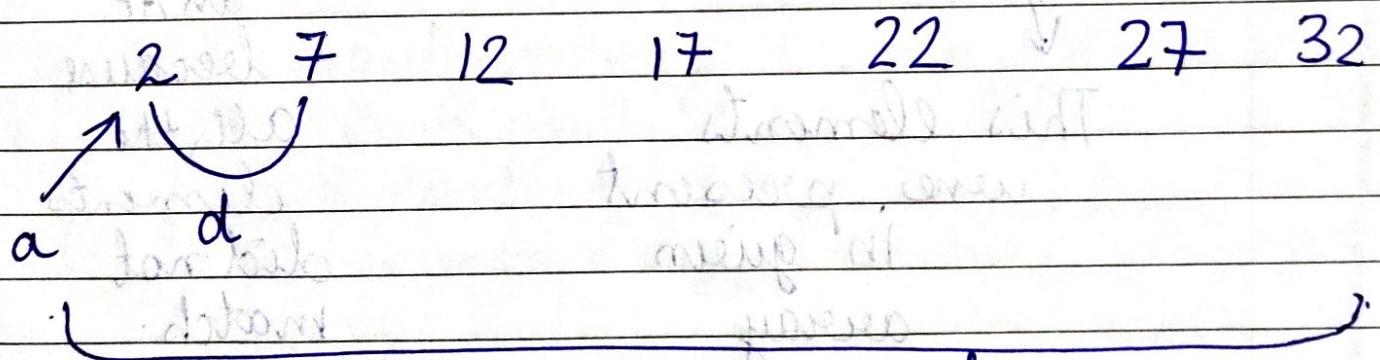
find all these 3
 $a = 2$
 $d = 5$.
length of given array = 7

(d) Create your own AP Sequence using.

$$a = 2$$

$$d = 5$$

$$\text{length} = 7$$



Check whether all these digits are there in HashSet or not

(e) Now, we need to go through this AP array & check whether the digits are present or not

Created
by us

2 7 12 17 22 27 32

HashSet ↴

2

7

12

17

22

27

37

32

16

not
present

So
the
given
array

is not
an AP
because

all the
elements
did not
match.

This elements
were present
in given
array.

Way 1 → Sorting the Array & then check ap
 $\hookrightarrow O(n \log n)$

Way 2) → Using Hashset & then check ap
 $(O(n))$

<code>

class Solution

{ boolean checkAP (int arr[], int n)

{ HashSet < Integer > set = new

Date _____
Page No. _____

HashSet < Integer >

}

int min = Integer . MAX_VALUE;

int secondMin = Integer . MAX_VALUE;

Initially
min
at value + no
diff

for (int i = 0; i < n; i++)

{ if (arr[i] < min)

{ secondMin = min;
min = arr[i];

O(n)

else if (arr[i] < secondMin)

{ secondMin = arr[i];

set.add (arr[i]);

int diff = secondMin - min;

for (int i = 0; i < n; i++)

{ if (set.contains (min))

{ min = min + diff; }

(n)

else

{ return false; }

}

return true;

b3

The last

array

at min,
second min की

value find की

array की रखी

values को set में

START

min & diff

add करते जाते

to create

AP sequence

& check if

it in array

or not

size of AP

values AP के लिए नहीं

simply return false

Ques. 5 Array of doubled Pairs

-18 7 -8 6 14 3 34 -4 17 3 -9 6

Ex element किसी pair में होना चाहिए

या तो वो किसी value के double होना

या उसका double array में present होना.

Approach

1) frequency map बनाओ। (first Sort the array & then create

-18 → 1
7 → 1
-8 → 1.
6 → 1.2

14 → 1

3 → 1.2

34 → 1

-4 → 1

17 → 1

-9 → 1.

— —

2). अब फ़ुरे array पर traverse करो और

Ex value के लिए check करो आज उसका double
मौजूद हो तो value और उसके double की
frequency में 1 subtract करो
आज 0 frequency हो जाये तो Hashmap में से remove करो
hashmap पूरी empty हो जाये तो इस return करो otherwise false.

Cg → 1

Date _____
Page No. _____

Special
Case

Valid

2 4 0 0 8 1

Why
Sorting
is necessary

(2, 4, 0, 0, 8, 1)

We didn't

→ 2 Initially

Sort

2 → 1

4 → 1

0 → 2

2 check करता है

2 और $(2 * 2) 4$

HashMap में हैं

8 → 1

1 → 1

∴ Yes (2, 4) HashMap से remove हो जाएगी

→ 0 will make pair with 0.

∴ (0, 0) Hash map से remove हो जाएगी.

→ 8 does not have its double.

→ 1 also does not have its double.

map ↘

8 → 1

1 → 1

They are not removed
yet

So false will
be returned

Return false

wrong answer.

After Sorting

- ① Arrays.sort(arr).

0 0 1 2) 4 8

8

New traversal of array & create frequency Hashmap

0	\longrightarrow	2
1	\longrightarrow	1
2	\longrightarrow	1
4	\longrightarrow	1
8	\longrightarrow	1

0 has its double

③ 0 has its double
& will pair with 0 & 8
& 0 will be removed from map

④ 1 has its double

& will pair with 2
& remove (1, 2) from map.

⑤ 4 has its double & (4, 8) will be removed from map

⑥ New Hashmap is empty so return true

✓ correct answer

Ques. 5.

Rabbits in Forest

Date		
Page No.		

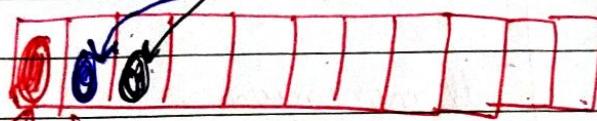
धो रक्क array queen होगा

a)

[10, 10, 10]

मतलब इसने 3 rabbits से लात की है

rabbits
कहिए हैं
मेरे जैसे 10 और 2



11 Total rabbit

मतलब 3 राब्बिस इसके जैसे 10 और 2 हैं

(i) $\text{Total rabbit जाताहो} = \text{Min} \rightarrow 11$ हो देती

b)

[2, 2, 3, 4, 2, 2, 3, 4, 3, 3]

2 → 5

3 → 4

4 → 2

प्रेरणा से 2 और
बिल्कुल वाले 5 Rabbit हैं

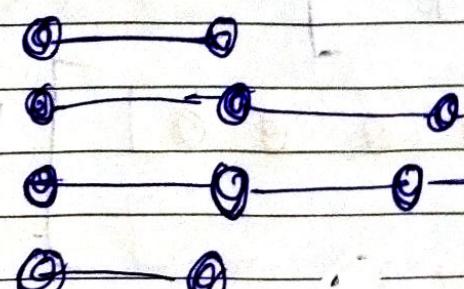
[2, 2, 2] 3

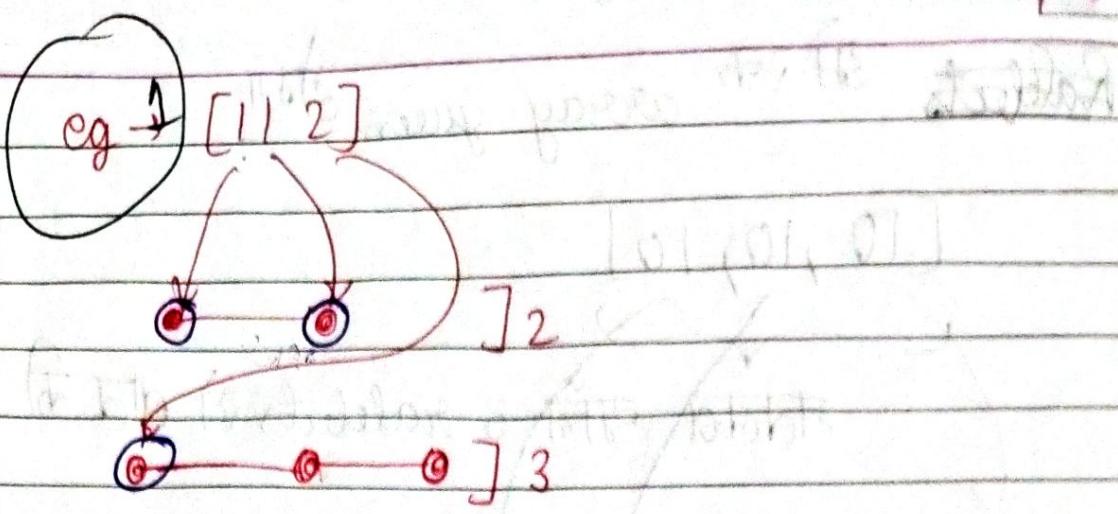
[2, 2]. } 3

[3, 3, 3, 3] 3

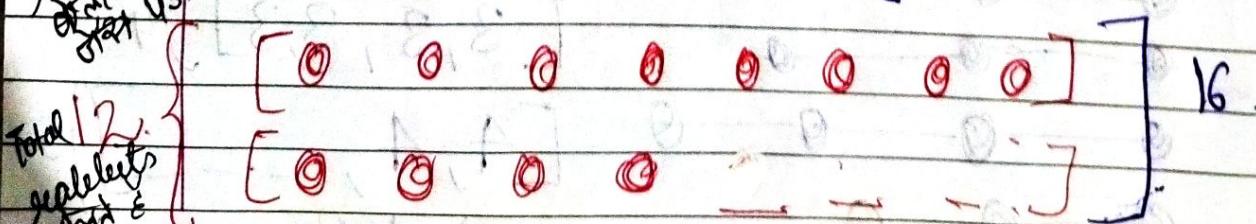
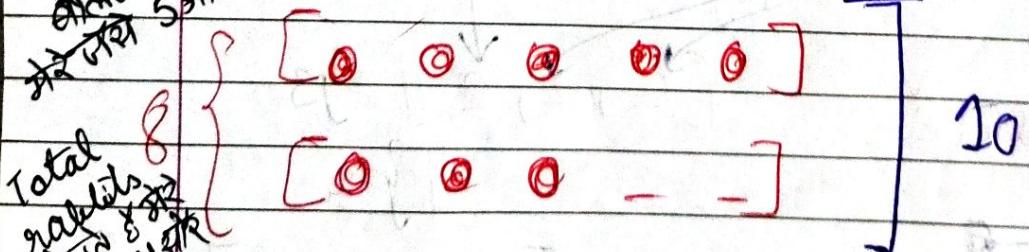
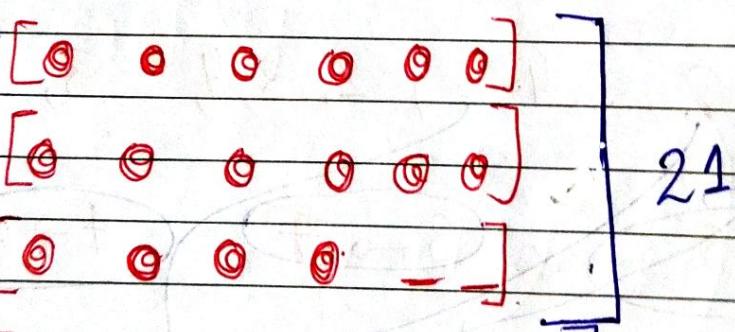
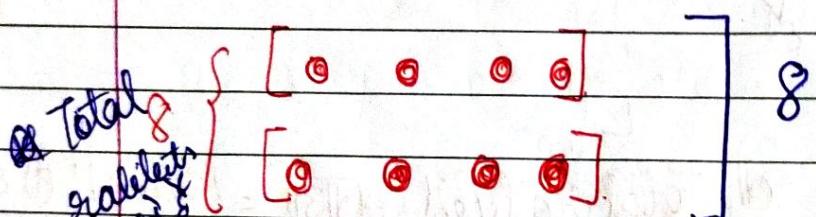
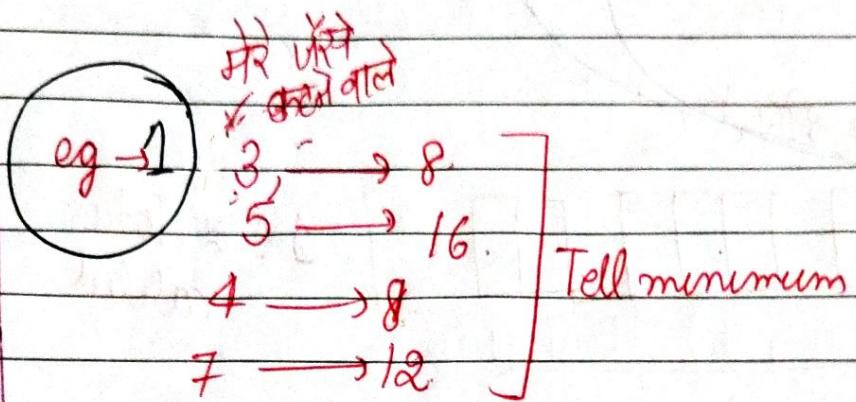
[4, 4, - - -] 5

Total = 15. नोट दृष्टि





Total minimum = 5.



$$\text{Total minimum} = 16 + 10 + 21 + 8 = 26 + 29 = 55$$

$cg \rightarrow 3$

$$2 \rightarrow 5$$

$$3 \rightarrow 6$$

$$1 \rightarrow 2$$

Date		
Page No.		

groups

group size

groups

group size

$$\left[\begin{array}{l} 6 \\ 4 \end{array} \right] \text{ceil} \times 4$$

$$(1 \cdot 5) \text{ceil} \times 4$$

$$2 \times 4$$

8 Rabbits

$$\left[\frac{2}{5} \right] \text{ceil} \times 5$$

$$\text{ceil}(0.4) \times 5$$

$$1 \times 5$$

5 Rabbits

$$\text{Total} = 8 + 5 + 6 = 8 + 11 = 19 \text{ Rabbit}$$

Approach

① [3, 4, 2, 3, 4, 2, 3, 2, 3, 2, 3, 2, 3]

↑

This is the given array.

② Now, create a frequency map.

$$2 \rightarrow 5$$

$$3 \rightarrow 6$$

$$4 \rightarrow 2$$

③ Now create an answer variable.
initially it is zero.

→ Now put a for loop on the key
→ get freq of key.

$$\text{ans} += \left(\frac{\text{freq}}{(\text{key} + 1) \text{ceil}} \right) * (\text{key} + 1)$$

$$\text{ans} += \left(\frac{\text{math.floor}(\text{freq} * 100)}{\text{ceil} * (\text{key} + 1) * 100} \right) * \text{key} + 1$$

↓ double

<code>

```
public static int solution (int[] arr)
{
    Map<Integer, Integer> map = new
        HashMap<>();
    for (int val : arr) {
        map.put(val, map.getOrDefault(val, 0) + 1);
    }
    int ans = 0;
    for (int key : map.keySet()) {
        int groupSize = key + 1;
        int repSizeOfSameSizeGroup = map.get(key);
        int noOfGroups = (int) Math.ceil((repSizeOfSameSizeGroup * 1.0) / groupSize);
        ans += (noOfGroups * groupSize);
    }
    return ans;
}
```

frequency map

Initially ans 0

groupSize

repSizeOfSameSizeGroup

Math.ceil works on double & return double

RateLimit