Date: 15th January 2022

Agenda →

1. Check arithmatic Seq.
2. Rabbits in the forest
3. Recurring seq. in fac.
4. Equivalent Subarrays
5. Pair with equal sum

1. Check arithmatic sequence :→

array → 1 (0) 9 (1) 25 (2) 33 (3) 5 (4) 13 (5) 21 (6) 17 (7) 29 (8)

Min. Number ↓ Max. Number

arithmatic seq → a_0 | a_0+d | a_0+2d | a_0+3d | ---- | $a_0+(n-1)d$

For arithmatic seq.

① a_0 → First term
② d → common diff.
③ n → no. of terms.

Given → Array, Random order.

Max. Number = Min. Number + (n-1)d (n → No. of Elements in an array.)

(n-1) d = Max. Number - Min. Number.

$$\Rightarrow \boxed{d = \frac{max - min}{n-1}}$$

After comon diff.

we can verify is array is AP or not.

~~*~~ when we are find max and min add Elements in HashSet

max = 33 , min → 1 , n = 9

$$d = \frac{max - min}{n - 1} = \frac{33 - 1}{9 - 1} = \frac{32}{8} = 4$$

d = 4

```java
public static boolean checkArithMaticSeq(int[] arr)
    // find min and max and add element in hashset
    if(arr.length <= 1) return true;
    HashSet<Integer> set = new HashSet<>();
    int min = Integer.MAX_VALUE;
    int max = Integer.MIN_VALUE;

    for(int val : arr) {
        min = Math.min(min, val);
        max = Math.max(max, val);
        set.add(val);
    }

    int n = arr.length;
    int d = (max - min) / (n - 1);
    int sum = min;
    while(sum < max) {
        sum += d;
        if(set.contains(sum) == false) return false;
    }
    return true;
}
```

Rabbits in the forest:

arr →

0	1	2	3	4	5	6	7	8
2	2	2	1	0	2	2	2	1

Rabbits
- Color / Appearance
  - Report
  - x Report

Min - no. of Rabits -

arr[i] → i^th Rabit, Similar Rabbits present in forest is arr[i]

a → 0 1 5 — ② → 3

b → 2 7 # # — ③ → 4

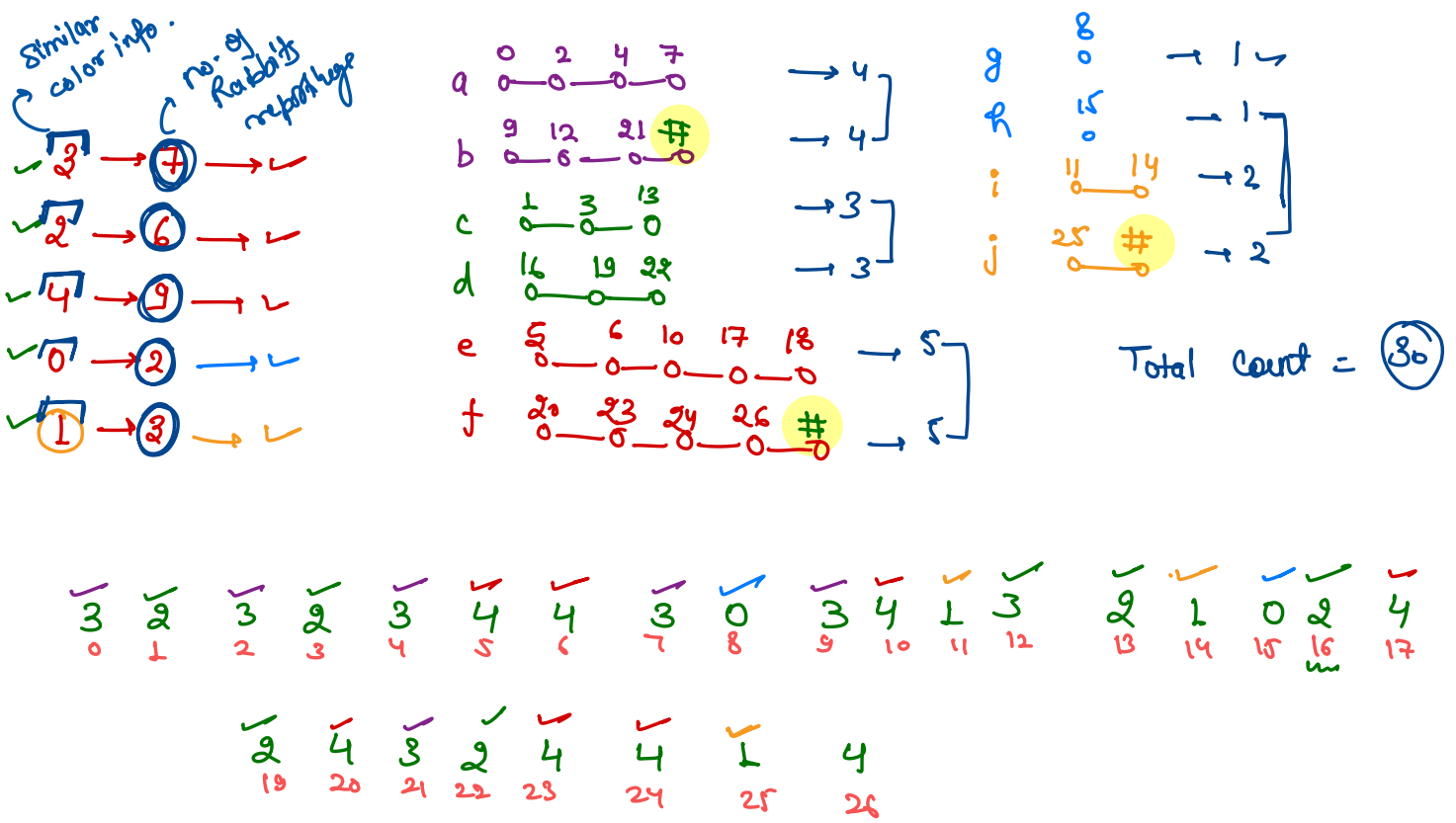
c → 3 8 — ① → 2

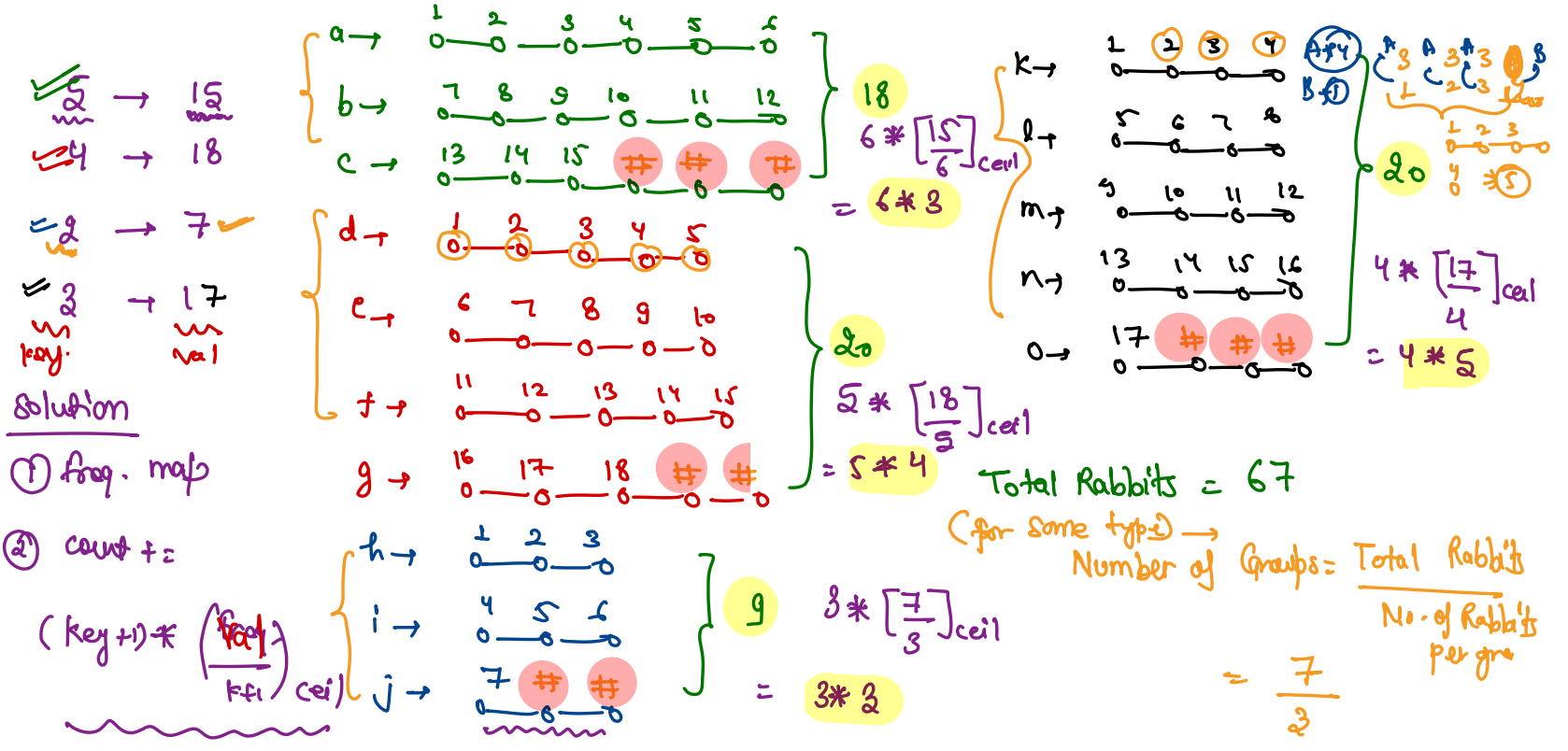
d → 4 — ① → 1

e → 6 # # — ② → 2

new grouping.

Total Rabbits - (13) Rabbits


Similar color info.
no. of Rabbits reporting age
a
b
c
d
e
f
g
h
i
j
Total count = 30


2 → 15
4 → 18
2 → 7
2 → 17
key
val
Solution
① freq. map
② count +=
(key+1) * (fq / kfi) ceil
a → 1 2 3 4 5 6
b → 7 8 9 10 11 12
c → 13 14 15 # # #
18
6 * [15/6] ceil
= 6 * 3
d → 1 2 3 4 5
e → 6 7 8 9 10
f → 11 12 13 14 15
g → 16 17 18 # #
20
5 * [18/5] ceil
= 5 * 4
h → 1 2 3
i → 4 5 6
j → 7 # #
9
3 * [7/3] ceil
= 3 * 3
k → 1 2 3 4
l → 5 6 7 8
m → 9 10 11 12
n → 13 14 15 16
o → 17 # # #
20
4 * [17/4] ceil
= 4 * 5
A+4
B+1
Total Rabbits = 67
(for some type) → Number of Groups = Total Rabbits / No. of Rabbits per grp
= 7/3
= 2.333 ≈ 3 Groups
Total Rabbits (Collective) = No of Rabbit per grp * Total Groups
= 3 * 3 = 9

Total person = ?

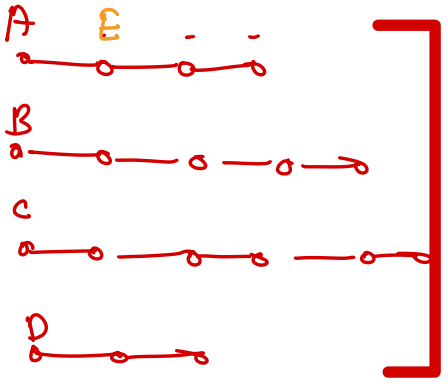
A → 3

B → 4

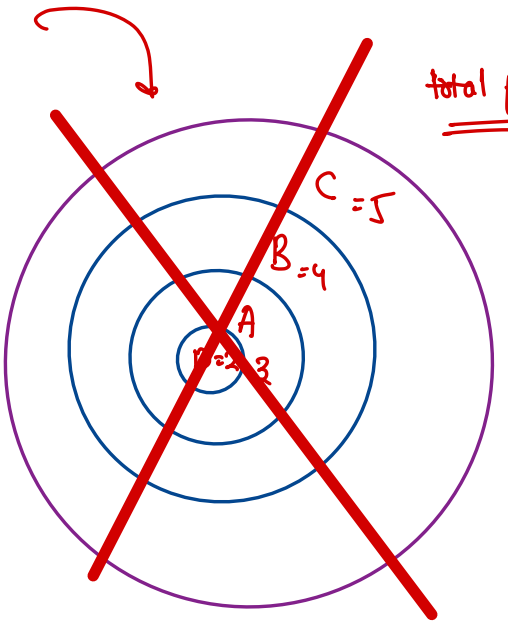
C → 5

D → 2 why not 5

E → 3



Grouping



total person = 5

groups of 3

Recurring Sequence in A Fraction

$\frac{8}{4}$ = 2

$\frac{37}{2}$ = 18.5

$\frac{428}{125}$ = 3.424

$\frac{14}{3}$ = 4.(6)

$\frac{47}{18}$ = 2.6(1)

$\frac{93}{7}$ = 13.(285714)

$4\overline{)8}$ → 2, 8, 0

$2\overline{)37}$ → 18.5, 2, 17, 16, 10, 10, x

$125\overline{)428}$ → 3.424, 375, 530, 500, 300, 250, 500, 0

$3\overline{)14}$ → 4.666.., 12, 20, 18, 20, 18, 20

$18\overline{)47}$ → 2.611.., 36, 110, 108, 20, 18, 20, 18, 2

After decimal If Remainder is start Repeating then bracket is Ragring

$7\overline{)93}$ → 13.285714.., 7, 23, 21, 20, 14, 60, 56, 40, 35, 50, 49, 10, 7, 30, 28, 20

zero for decimal

$\frac{93}{7}$

String = ~~~~ . ~~~~

First part = second part

First part = $\frac{\text{Numerator}}{\text{Denominator}}$

$= \frac{93}{7} = 13$

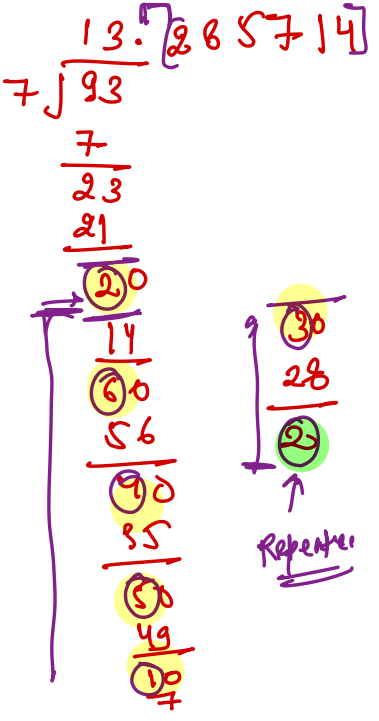
check if remainder is 0 or not

if remainder == 0 ⟶ ans ⇒ First part

else → (1) add decimal in answer

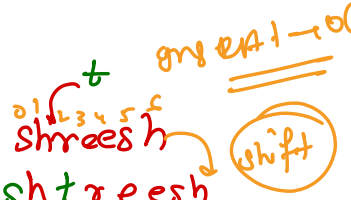
(2) change remainder to zero.

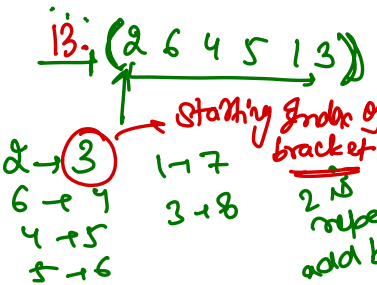
13.(2 6 4 5 1 3) ⟶ Result

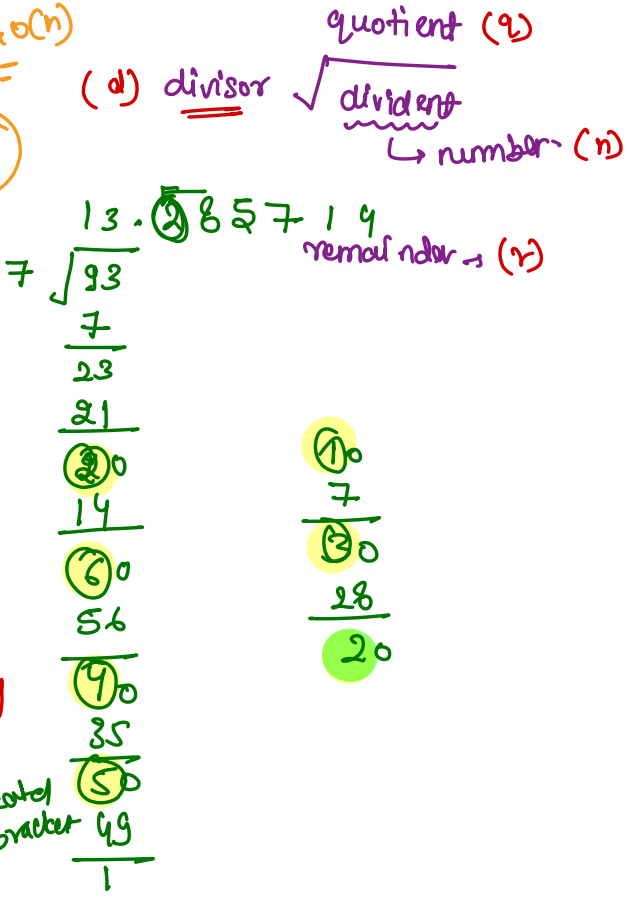


```
q = n/d; ⇒ 13
r = n%d; ⇒ 2
ans += q;
if(r == 0) {
    return ans;
}
ans += "."; // decimal ⇒ "13."
while(r != 0) {
    r *= 10
    q = r/d;
    r = r%d;
    ans += q;
}
```

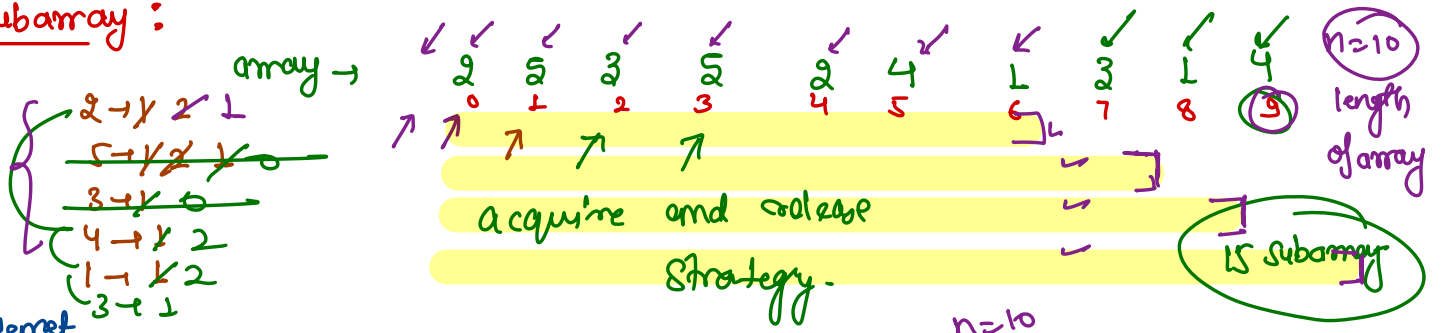
$q = \frac{2_0}{7}$







Equivalent Subarray :



distinct Element

① Find using HashSet = k → No. of distinct Element k=5

② No. of subarrays having k distinct Element

2
5
3
4
1
} k=5

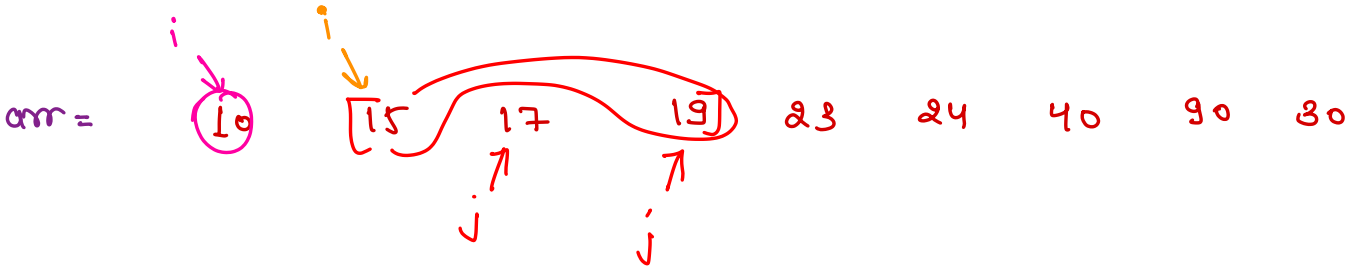
```
while(true)
  while( ) →
    acquire till size of hashmap is not k
    if (k==1) include in answer and move to ward release
  while( ).
  release
    add ans. until size of hashmap is maintain
    at k.
  if both loop is not visited
    → break
```

n=10
arr. length.

10-6 = 4
10-6 = 4
10-6 = 4
10-7 = 3

Pair with Equal Sum:

boolean → True / False.



A + B = C + D where, A, B, C and D are element of array.

Time complexity → $O(n^2)$

Space complexity → $O(n)$

25 → 10, 15

27 → 10, 17

29 → 10, 19

33 → 10, 23

34 → (10, 24), (15, 19)

50 → 10, 40

100 → 10, 90

40 → 10, 30

32 → 15, 17

if arr[i] + arr[j] → present in set

↳ return true