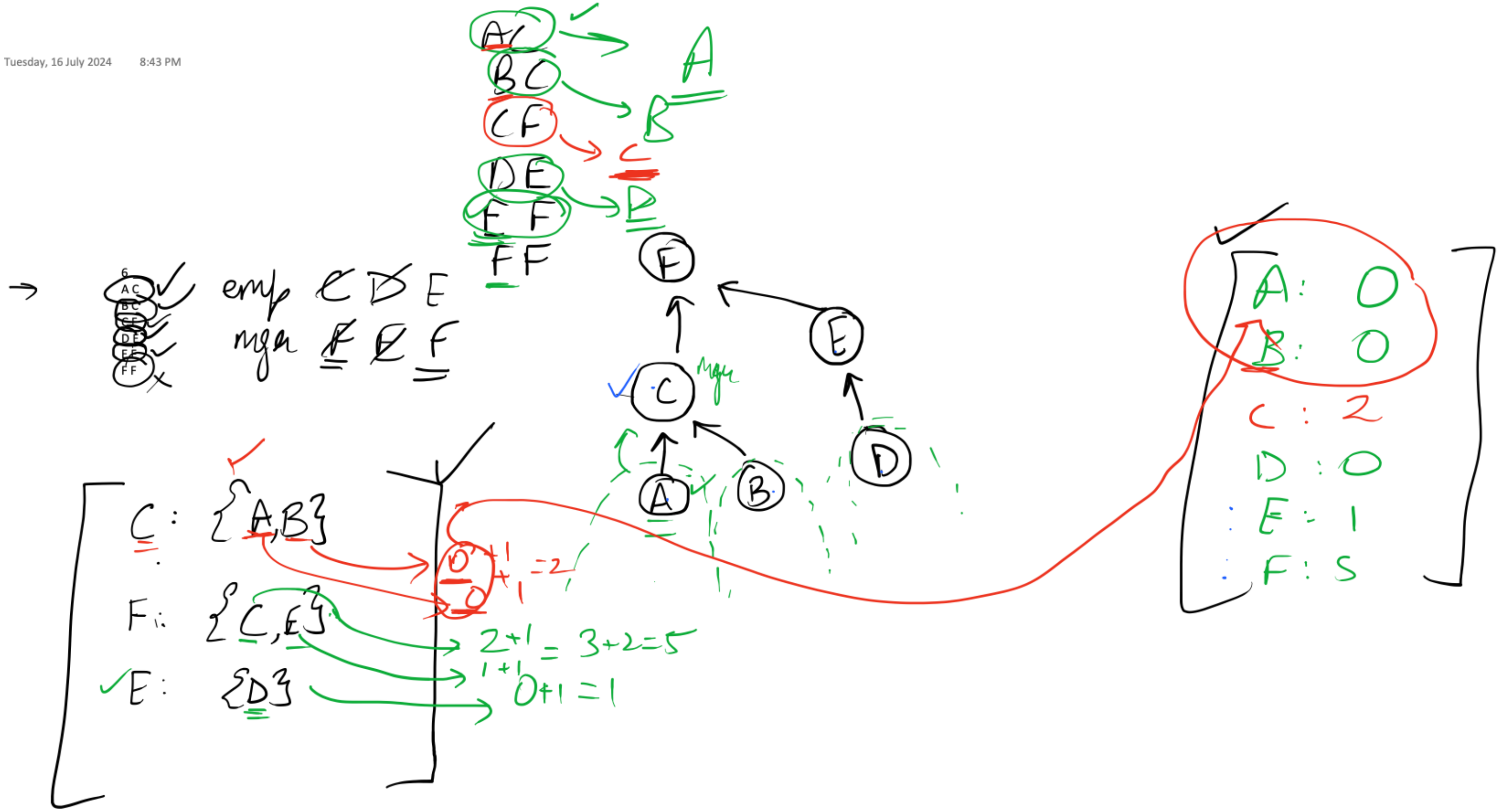
Tuesday, 16 July 2024 8:43 PM



C : {A,B}

F : {C,E}

E : {D}

0+1, 0+1 =2

2+1 = 3+2=5

1+1, 0+1 =1

A :	0
B :	0
C :	2
D :	0
E :	1
F :	5

calculate number will run just once per each emp.

[illegible]

Problem with Given Difference

6

$$\begin{array}{cccccc} & & & i & & j\ \checkmark \\ \rightarrow 5 & 10 & 3 & 2 & 50 & 80 \end{array}$$

Diff → $\underline{\underline{78}}$ →

↳ Return $\underline{\underline{1}}$

Approaches

1) Using 2 loops check for every pair

↳ $\underline{\underline{O(n^2)}}$

2) $\underline{\underline{\text{Sort}}}$ + B.S ⇒ $O(n\log n)$

↳ $O(n\log n)$

$\underline{2}$, $\underline{3}$, $\underline{5}$, 10, $\underline{50}$, $\underline{8'}$ $\underline{70}$

↑

BS to find $\underline{\underline{X}} = 2 + \text{diff} = 2+78 = 80$

↓ ↳ $\text{diff} = \underline{\underline{X-2}}$

$n \times \log n$

✓2) Set ⇒ $O(n)$

↳ Add to the set & compare

We can keep a track visited numbers ↑

{ 2, 3, 5, 10, 50, }

↳ Search, add → O(1)

Diff

3 → 3 + 78 = 81

3 − 78 = −75

5 → 5+78 = 83×

↳ 5−78 = −73×

10 → 10+78 = 88×

↳ 10−78 = −68×

80 → 80+78 = 158×

↳ 80−78 = $\boxed{2}$

50 +78 = 128

50 −78 = −28

1st 2nd

↓ ↓

$\boxed{X - Y}$ = Diff

X = Diff + Y = Diff + arr[i]

Y = X − Diff ⇒ 1R' = Arr[i] − Diff

Array Pairs divisible by K

10, 5

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

(2,3) (4,6) (5,10) (7,8) (1,9)

× (2,8) (1,4) (3,7) (5,10) (6,9)

1 % 5 = 1

2 % 5 = 2

3 % 5 = 3

4 % 5 = 4

5 % 5 = 0

6 % 5 = 1

7%

8%

9

10

1 : 2

2 : 2

3 : 2

0

↘ 4 : 2 ↳

↳ Sum of pair should be divisible by K

(K - key)

1 → 4 ?, 9 ?

↳ 5 - 1 ⇒ 4

5 ⇒ 4 + 1

↳ Dividend = Divosor × Quotient + Remainder

9 = 5 × 1 + 4

4 = 5 × 0 + 4

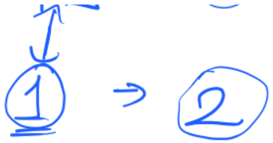
1 = 5 × 0 + 1

6 = 5 × 1 + 1

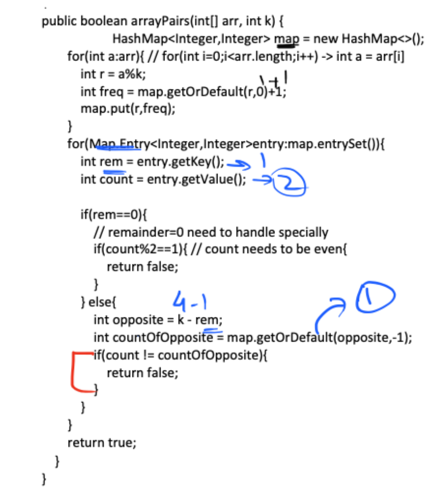
↳ 4 → 2

5 - 1

(9, 6), (4, 1)



(9,1) , (4,6)



```
public boolean arrayPairs(int[] arr, int k) {
            HashMap<Integer,Integer> map = new HashMap<>();
    for(int a:arr){ // for(int i=0;i<arr.length;i++) -> int a = arr[i]
      int r = a%k;
      int freq = map.getOrDefault(r,0)+1;
      map.put(r,freq);
    }
    for(Map.Entry<Integer,Integer>entry:map.entrySet()){
      int rem = entry.getKey();
      int count = entry.getValue();

      if(rem==0){
        // remainder=0 need to handle specially
        if(count%2==1){ // count needs to be even{
          return false;
        }
      } else{
        int opposite = k - rem;
        int countOfOpposite = map.getOrDefault(opposite,-1);
        if(count != countOfOpposite){
          return false;
        }
      }
    }
    return true;
  }
}
```

1+1 → 1 → 2 4-1 → 1

5 10
1 2 3 4 5 6

3 3
1, 2, 3, 4, | 5, 6

K= 4

1 : ~~1~~2
2 : ~~1~~ ~~2~~2
3 : 1
0 : 1

~~k=~~2

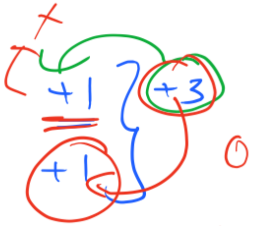
~~r=4%~~4

r= 5%4
= 1

r=6%4
= 2

1 = 4×0+1 +3

5=4×1+1 +3



Only 1 is possible

Largest Subarray with Sum0

15, -2, 2, -8, 1, 7, 10, 23

1) To use 2 loops and generate all the possible subarrays.

↳ If any subarray is found with 0 sum

↳ store it's length

↳ return the maximum length

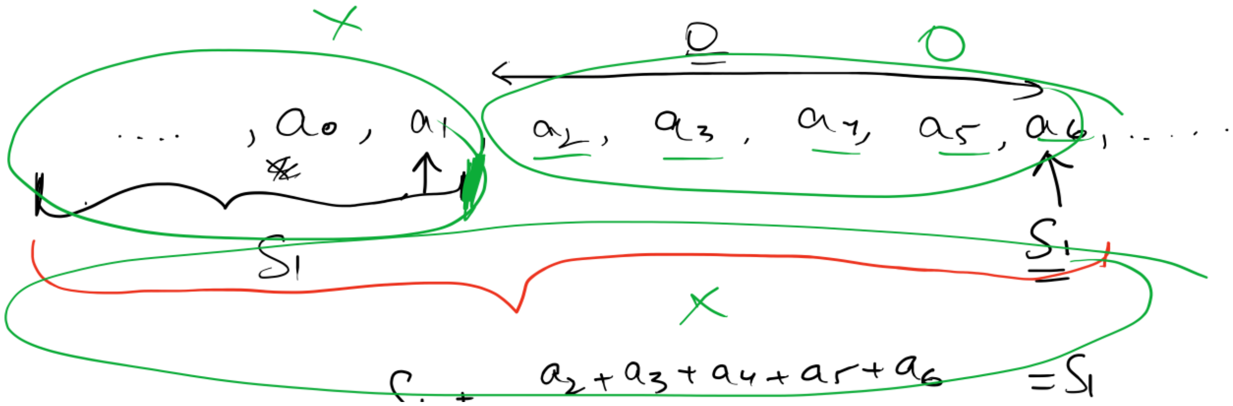
$\underline{\underline{O(n^2)}}$

2) ~~Sort~~ + ? → Subarray

-8, -2, 1, 2, 7, 10, 15, 23

$X + 0 = X$

15, -2, 2, -8, 1, 7, 10, 23



$$\hookrightarrow \quad a_2 + a_3 + a_4 + a_5 + a_6 = 0$$

$$\begin{array}{cccccccc} \times & & & = 0 & & & \times & \\ 15, & -2, & 2, & -8, & 1, & 7, & 10, & 23 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \quad ✓$$

$$\text{Sum} = \cancel{0}\ 15 \underset{(1)}{-2} = 13 \underset{(2)}{+2} = 15 - 8 = 7 + 1 = 8 + 7$$

$$= 15 + 10$$

$$= 25$$

$$23$$

$$48$$

$$\text{len} = \cancel{0}$$

$$2, \quad 5 - 0 \quad = 5$$

Map <Sum, Index>

0	-1
15	0
13	1
7	3
8	4
25	6
48	7

$$\underset{0}{[2,} \ \underset{1}{-2]} \quad ✓$$

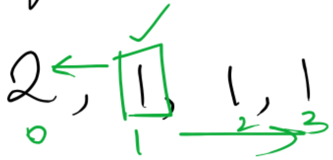
$$\text{sum} = 2 - 2 = 0$$

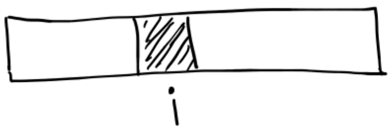
$$\text{len} = (0, \ 1 - (-1))$$

$$= (0, 2)$$

0	-1
2	0

Equilibrium Index





1) Using 2 loops

i →

j = 0 → i-1

k = i+1 → n-1)

2 → Sum of [0 → i-1] → Prefix Sum

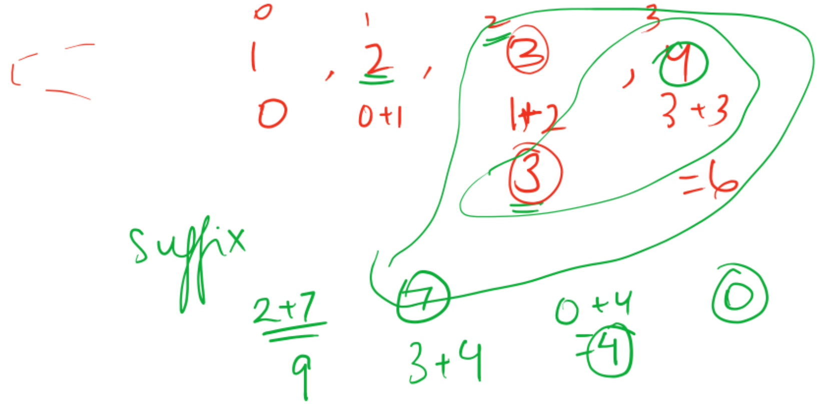
↳ Sum [i+1, n-1]

↳ Suffix Sum

2) Using Prefix Sum & Suffix Sum

Prefix Sum → PS[i] = PS[i-1] + arr[i-1]

↳ Calculating the previous sum in O(1)


0
1
2
3
1
2
3
4
0
0+1
1+2
3
3+3
=6
suffix
2+7
9
7
3+4
0+4
4
0