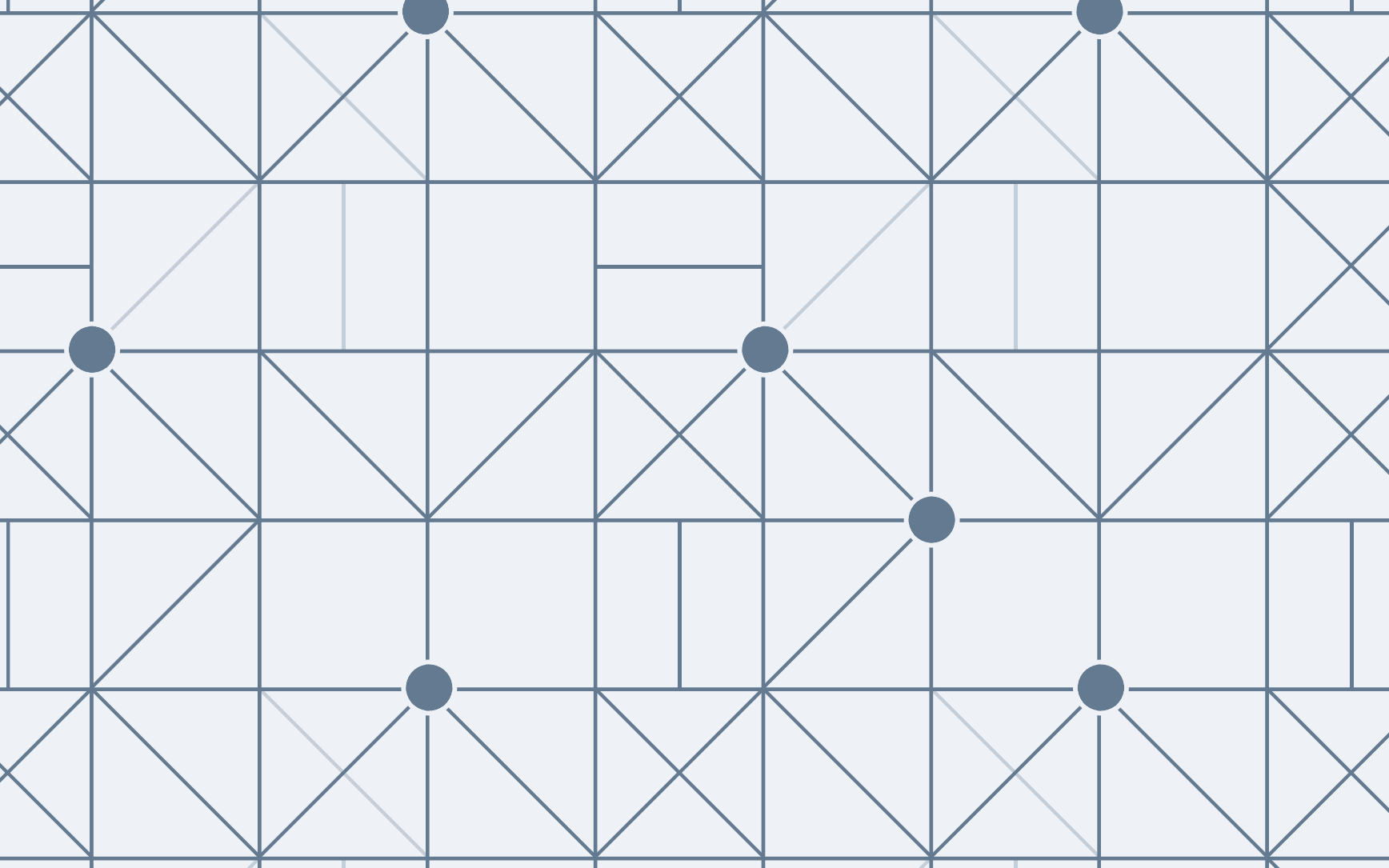
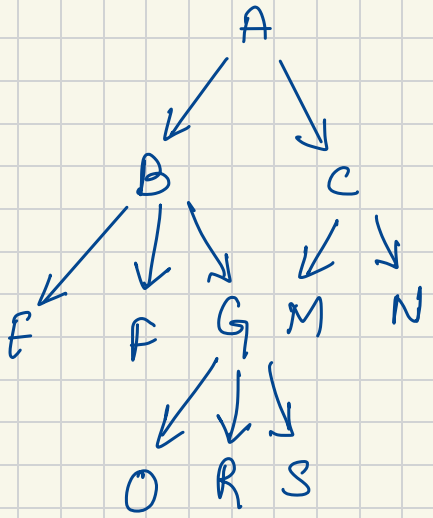


Agenda

- → Employees and Manager
- → few que from Assignment

Employee And Manager.



E → 0
F → 0
O → 0
R → 0
S → 0
M → 0
N → 0
B → 6

G → 3
C → 2
A → 10

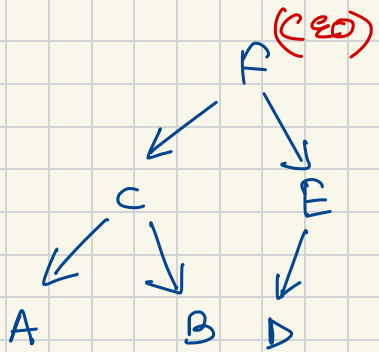
O/P

Input

```
6
A C
B C
C F
D E
E F
F F
```

CEO

Emp (Key) Mngr (Value)



A → 0

B → 0

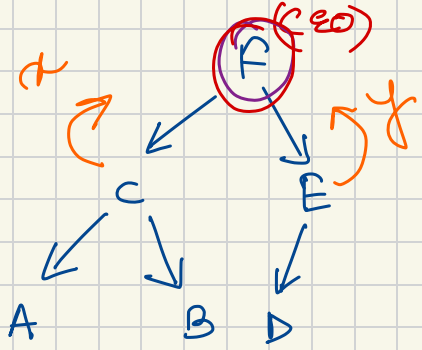
D → 0

C → 2

E → 1

F → 5

O/P



$(x+1) + (y+1)$ = total Emp under F

key → value

F → {C, E}

C → {A, B}

E → {D}

direct reportees

```
getNumberOfEmpUnderMe (char mngr)
{
    total = 0;
    → go to all the direct reportees and get
      number of emp under them
      total += (x + 1)
               → for him
    HashMap {mngr, total}
}
```

```java
for (String employee : emp.keySet()) {
    String manager = emp.get(employee);

    if (manager.equlas(employee) == true) {
        // term him/her as CEO
        CEO = employee;
    } else {
        ArrayList<String> directs = directReportees.getOrDefault(manager, new ArrayList<String>());
        directs.add(employee);
        directReportees.put(manager, directs);
    }
}
```

D

E

Input

```
6
A C
B C
C F
D E
E F
F F
```

Direct Reportees

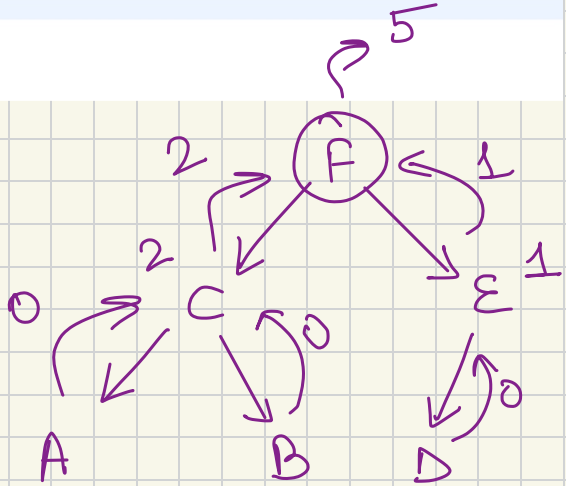
Mngr	directs
C	{A, B}
F	{C, E}
E	{D}

```java
// Faith: returns number of a employees under a manager
public int getAllEmpUnderMe(String mngr, HashMap<String, ArrayList<String> directReportees, HashM
    if (directReportees.containsKey(mngr) == false) {
        map.put(mngr, 0);
        return 0;
    }

    int totalEmployees = 0;

    for (String direct : directReportees.get(mngr)) {
        int employeesUnderYou = getAllEmpUnderMe(direct, directReportees, map);
        totalEmployees += (employeesUnderYou + 1)
    }

    map.put(mngr, totalEmployees);

    return totalEmployees;
}
```

F

Mngr	directs
C	{A, B}
F	{C, E}
E	{D}

A → 0
B → 0
C → 2
D → 0
E → 1
F → 5



Problem with a given difference

arr[] = {5, 10, 3, 2, 50, 80} B = 78

Brute force

```
for(int i=0 ⟶ n)
{
    for(j=i+1 ⟶ n)
    {
        if(arr[i] - arr[j] == B)
            return true;
        else if(arr[j] - arr[i] == B)
            return true;
```

TC: $O(N^2)$
SC: $O(1)$

arr[] = {5, 10, 3, 2, 50, 80} B = 78

pair(x, y)

$x - y = B$ } $x = B + y$ }

$y - x = B$ } $x = y - B$ }

$y = 5$

$x = 83$

$x = -73$

$y = 10$

$x = 88$

$x = -68$

Hashset

5, 10, 3, 2, 50

potential x value

$y = 3$

$x = 81$

$x = -75$

$y = 2$

$x = 80$

$x = -76$

$y = 50$

$x = 128$

$x = -28$

$y = 80$

$x = 158$

$x = 2$

True (2, 80)

arr[] = { 5, 10, 3, 2, 50, 80 } B = 45

sort

{ 2, 3, 5, 10, 50, 80 }

Si

ci

X Can't Be Solved

78

Array Pair Divisible By K

arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} K = 5

even sized array

size of the array

(N/2) such pair, where sum of each pair is divisible by K

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

10 10 10 10 15

→ is it possible or not?

arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} K = 5

pair (x, y)

$x + y = \alpha K$ {some 'α' multiple of K}

$K \overline{)x} \; q$

r

$x = K \times q + r$

$$(q_1 \times k + r_1) + (q_2 \times k + r_2) = \alpha k$$

$$(q_1 + q_2)k + (r_1 + r_2) = \alpha k$$

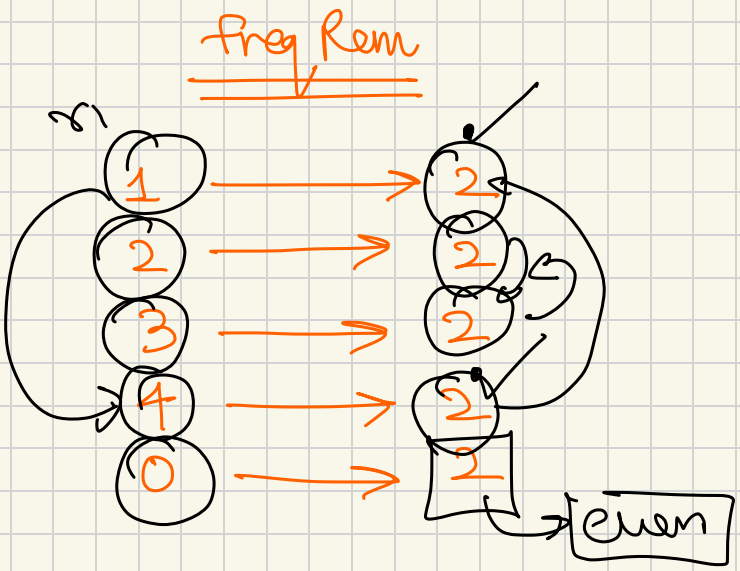
divisible by k

should be divisible by k

$(r_1 + r_2) = k$ or $(r_1 + r_2) = 0$

arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} K = 5

rem: 1 2 3 4 0 1 2 3 4 0



$r_1 + r_2 = k$

$r_2 = k - r_1$

$5 - 1 = 4$

$5 - 2 = 3$

Largest Subarray with Sum Equal to zero

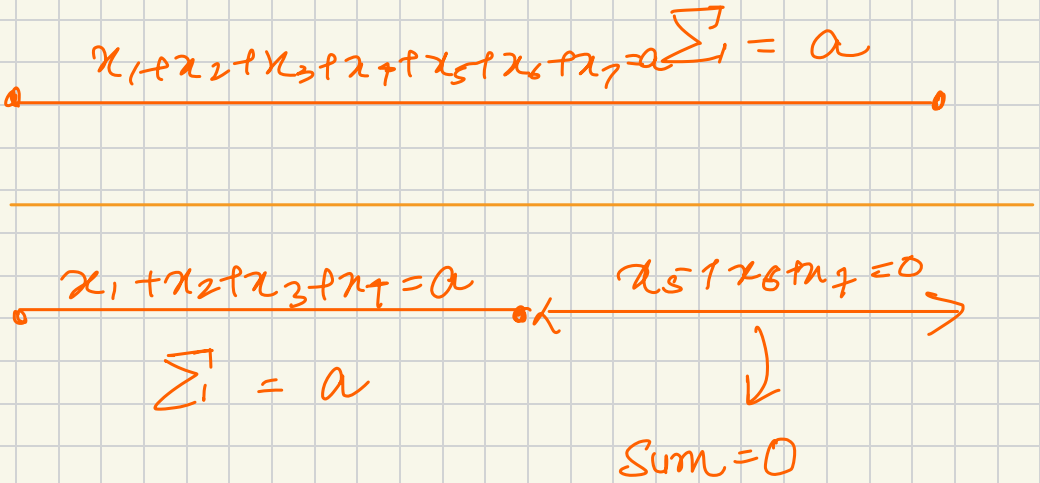
arr[] = {15, -2, 2, -8, 1, 7, 10, 23}

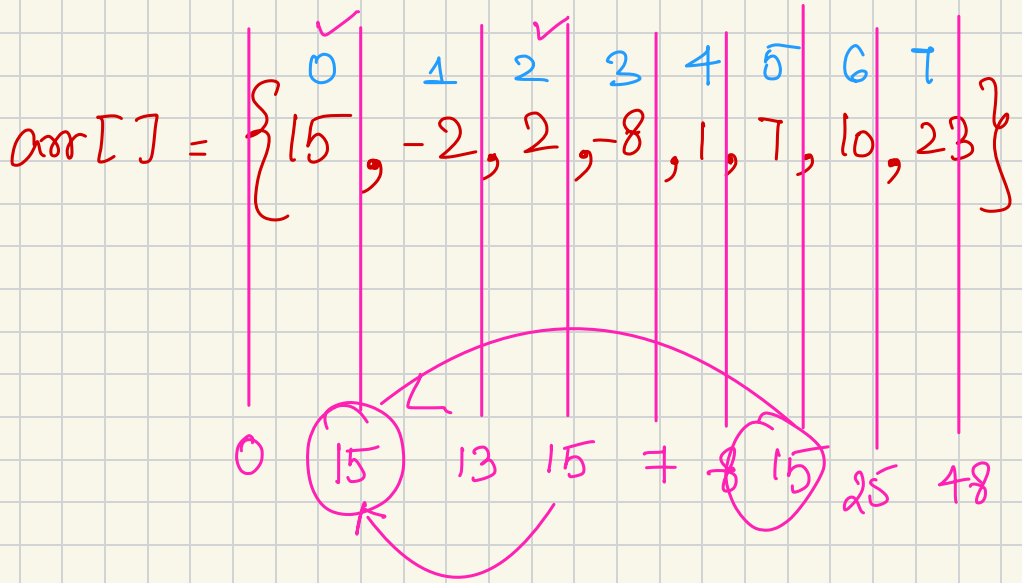
Bruteforce

↳ Calc. sum of each subarray, store largest length from them

TC: $O(N^2)$
SC: $O(1)$

$$arr[] = \{15, -2, 2, -8, 1, 7, 10, 23\}$$
(indices: 0 1 2 3 4 5 6 7)

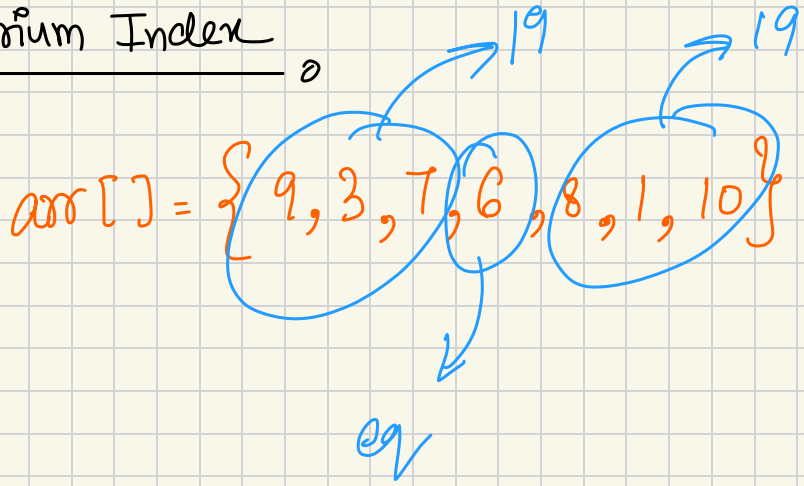




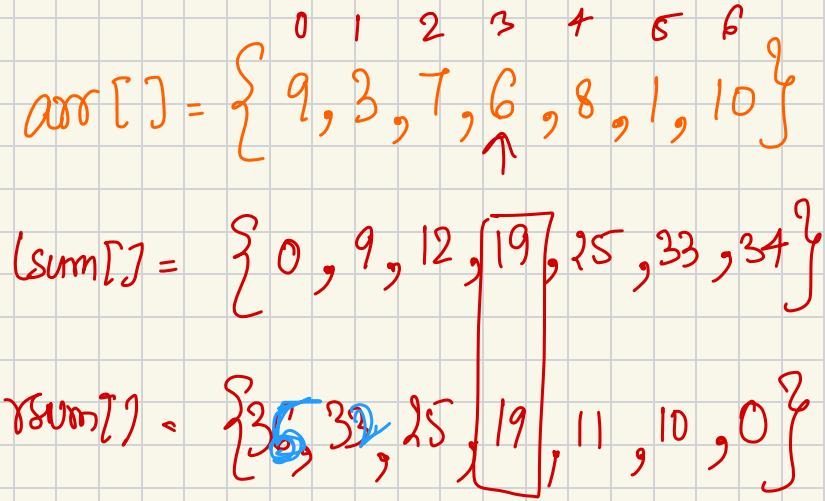
len = curIndex - firstSeenThatSum

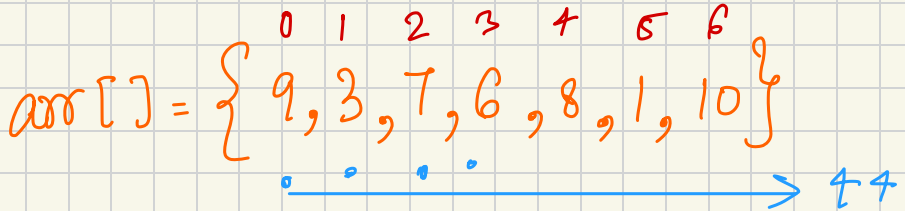
len = 5

Equilibrium Index



Equilibrium Index: index for whose sum of left people is equal to sum of right people


0 1 2 3 4 5 6
arr[] = { 9, 3, 7, 6, 8, 1, 10 }
lsum[] = { 0, 9, 12, 19, 25, 33, 34 }
rsum[] = { 25, 19, 11, 10, 0 }



lsum = ~~0~~ ~~9~~ ~~12~~ 19

lsum + rsum + arr[i] = totalSum

rsum = totalSum − lsum − arr[i]

44 − 0 − 9 = 35

44 − 3 − 9 = 32 ✓

44 − 12 − 7 = 25

44 − 19 − 6 = 19 ✓