



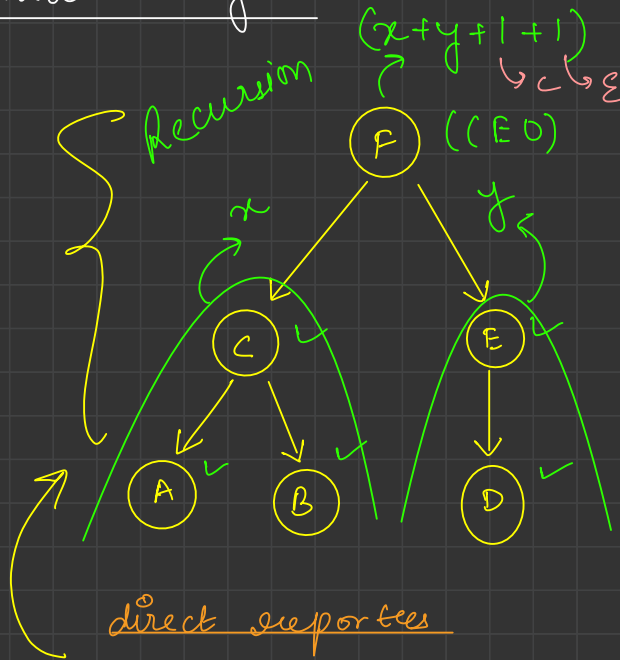
# Employees and Managers

eg

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

key value

Hash Map



O/P

A	→	0
B	→	0
C	→	2
D	→	0
E	→	1
F	→	5

{ faith tell me how many are under you }

✓ char , AL < char >

{

C → {A, B}

F → {C, E}

E → {D}

✓ CEO = F

```

// faith: returns number of people under mgr
public int cntOfEmployeeUnderMrg(HashMap<String, ArrayList<String>> directReport, String mgr, HashMap<String, Integer> underMe)
{
    ① if (directReport.containsKey(mgr) == false) {
        underMe.put(mgr, 0);
        return 0;
    }

    int cnt = 0;
    ② for (String emp : directReport.get(mgr)) {
        cnt += cntOfEmployeeUnderMrg(directReport, emp, underMe) + 1;
    }
    ③ underMe.put(mgr, cnt);
    return cnt;
}

```

direct Reportee

C → {A, B}  
 E → {D}  
 F → {C, E}

under Me

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

1  
 5



call stack

```

public void EmpUnderManager(Map<String, String> emp)
{
    // create a direct reportees hashmap
    HashMap<String, ArrayList<String>> directReport = new HashMap<>();

    String ceo = "";
    for (String employee : emp.keySet()) {
        String mgr = emp.get(employee);

        if (mgr.equals(employee)) {
            ceo = employee;
            continue;
        }

        if (directReport.containsKey(mgr) == true) {
            ArrayList<String> underThisMgr = directReport.get(mgr);
            underThisMgr.add(employee);
            directReport.put(mgr, underThisMgr);
        } else {
            ArrayList<String> underThisMgr = new ArrayList<>();
            underThisMgr.add(employee);
            directReport.put(mgr, underThisMgr);
        }
    }
}

```

{A, B}  
 {C, E}

emp

keys	values
6	
A	C
B	C
C	F
D	E
E	F
F	F

CEO = F

String	AL<String>
C	{A, B}
F	{C, E}
E	{D}

direct Report

## Problem with given diff

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

### Brute force

$\left\{ \begin{array}{l} TC: O(N^2) \\ SC: O(1) \end{array} \right.$

for (i → 0 → n)

for (j → i+1 → n)

if (arr[i] - arr[j] == B || arr[j] - arr[i] == B)  
return 1;

return 0



arr[]: { 5, 10, 3, 2, 50, 80 }

B = 78

TC:  $O(N)$   
SC:  $O(N)$  }  $(x, y) \leadsto \underline{\text{pair}}$

5, 10, 3, 2, 50

(HashSet)

$$x - y = B \quad \text{or} \quad y - x = B$$

$$\downarrow$$

$$y = x - B$$

$$\downarrow$$

$$y = x + B$$

$$\left\{ \begin{array}{l} x = 2 \\ y = 2 - 78 = -76 \\ y = 2 + 78 = 80 \end{array} \right.$$

$$x = 5$$

$$y = 5 - 78 = -73 \checkmark$$

$$y = 5 + 78 = 83 \checkmark$$

$$x = 10$$

$$y = 10 - 78 = -68$$

$$y = 10 + 78 = 88$$

$$x = 3$$

$$y = 3 - 78 = -75$$

$$y = 3 + 78 = 81$$

$$x = 50$$

$$y = 50 - 78 = -28$$

$$y = 50 + 78 = 128$$

$$x = 80$$

$$\checkmark y = 80 - 78 = 2 \checkmark$$

Ans true

# Array Pairs divisible by k

arr[] = { 0 1 2 3 4 5 6 7 8 9 }  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10



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

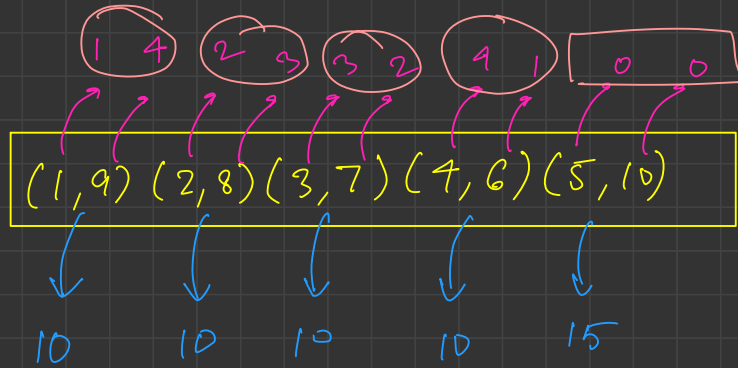
↓ ↓ ↓ ↓ ↓  
10 10 10 10 15

Even  
N=10 k=5 ✓  
N/2 = 5 pairs

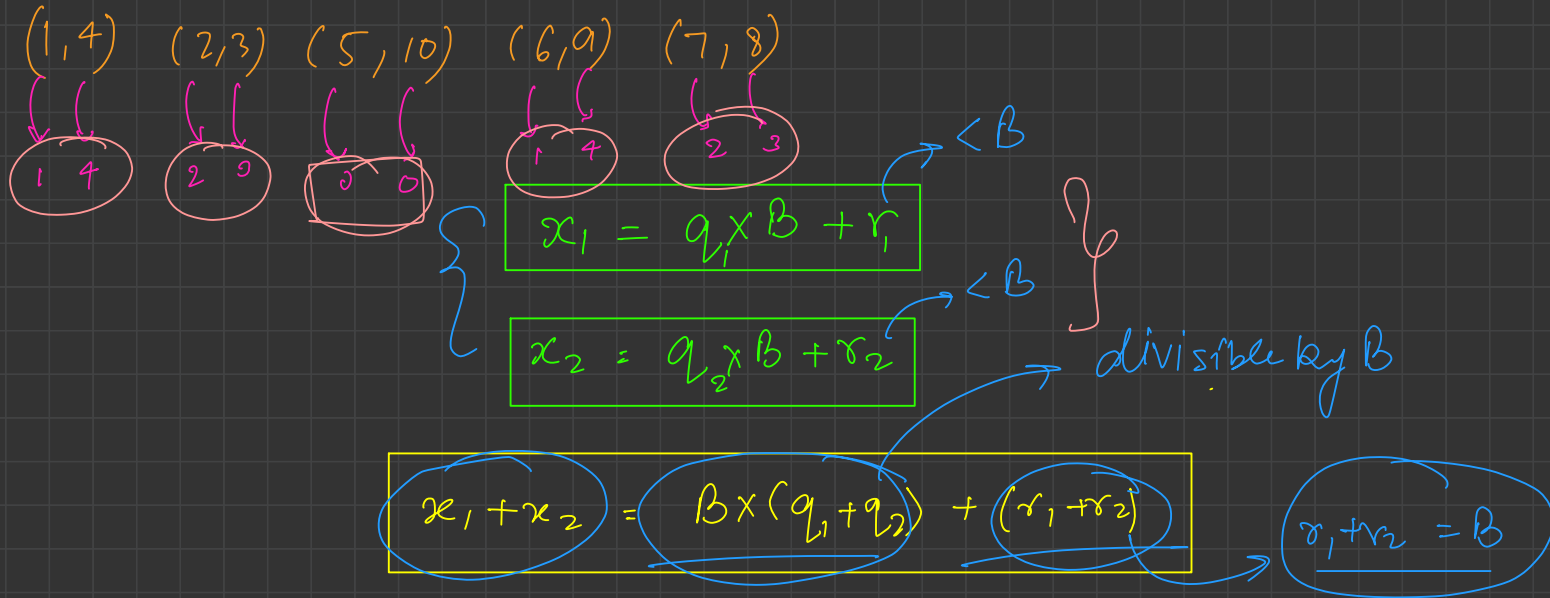
Yes!







$K=5$





largest Subarray with zero Sum.

arr[]: { 15, -2, 2, -8, 1, 7, 10, 25 }

Brute force

compute all subarrays with zero sum, and store maxLen

$$\begin{cases} TC: O(N^2) \\ SC: O(1) \end{cases}$$

arr[]: { 0 1 2 3 4 5 6 7 8  
~~15~~, ~~-2~~, ~~2~~, ~~-8~~, ~~1~~, ~~7~~, ~~10~~, ~~25~~, ~~-48~~ }

~~15~~ ~~13~~ ~~10~~ ~~8~~ ~~15~~ ~~25~~ ~~7~~ 0

0 → -1

15 → 0

13 → 1

7 → 3

8 → 4

25 → 6

48 → 7

maxLen = ~~7~~ 5

len = worldx - seenFirstTime

= 2 - 0 = 2

= 5 - 0 = 5