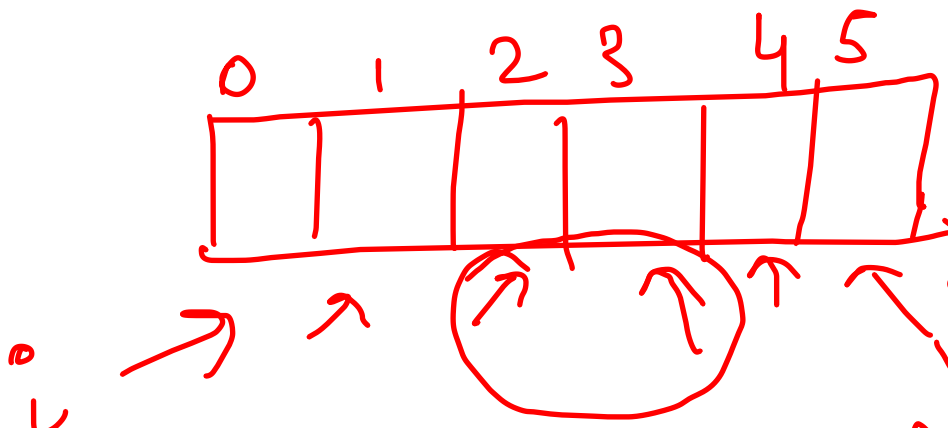
→ Two pointers

→ frequency array

Two pointers



$n^2 \rightarrow n$ — TC

i = 0, i + 1

j = n - 1, j - -

while (i < j)

i = 2, 3

j = 3, 2

HW_Rotate Number 1

```
    Scanner sc = new Scanner(System.in);
    String n = sc.nextLine();

    int k = sc.nextInt();

    String result = rotate(n,k);
    System.out.println(result);

}

public static String rotate(String n, int k){
    int len = n.length();

    k = k % len;

    if(k < 0){
        k = len + k;
    // convert left rotation to right
    }
    String s = n.substring(k) + n.substring(0,k);
    return s;
}
```

TC, SC

O(1)

"1234" → n len = 4

k = -2

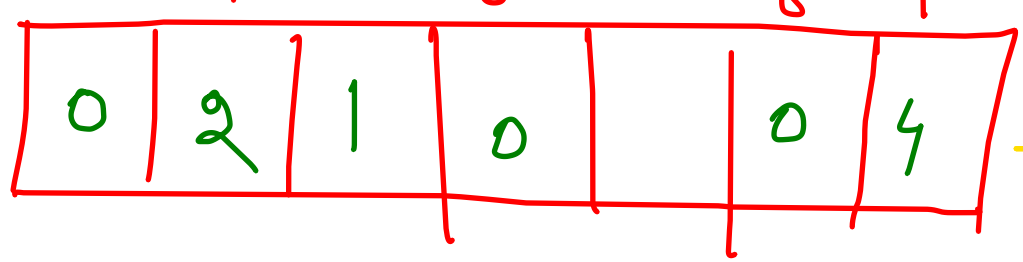
k = 4 + (-2) = 2

S = "34" + "12"

S = "3412" ✓

frequency array / Arrays as hashmap:

freq-[]



→ i (numbers)

→ freq[i] (occurence of i)

int → size 10

String → size 256 (if all chars included)

→ size 26 (if only alphabets)

TC → O(n)

SC → O(n)

arr → 10 → 10 freq.

→ 5 → 10

→ 100 → 10

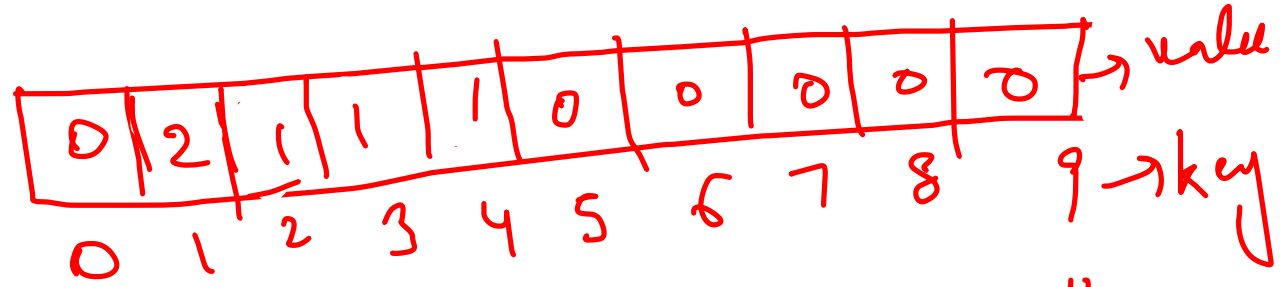
```java
Scanner s = new Scanner(System.in);
int n = s.nextInt();

int freq[] = new int[10];

// calculate freq of each no.
while(n > 0){
    int digit = n % 10;
    freq[digit]++;
    n /= 10;
}

// digit with max freq

int maxfreq = 0;
int maxfreqdigit = 0;
for(int i = 0; i < 10; i++){
    if(freq[i] > maxfreq){
        maxfreq = freq[i];
        maxfreqdigit = i;
    }
}
System.out.println(maxfreqdigit);
}
```

n = 11234

K V
0 → 0
1 → 2

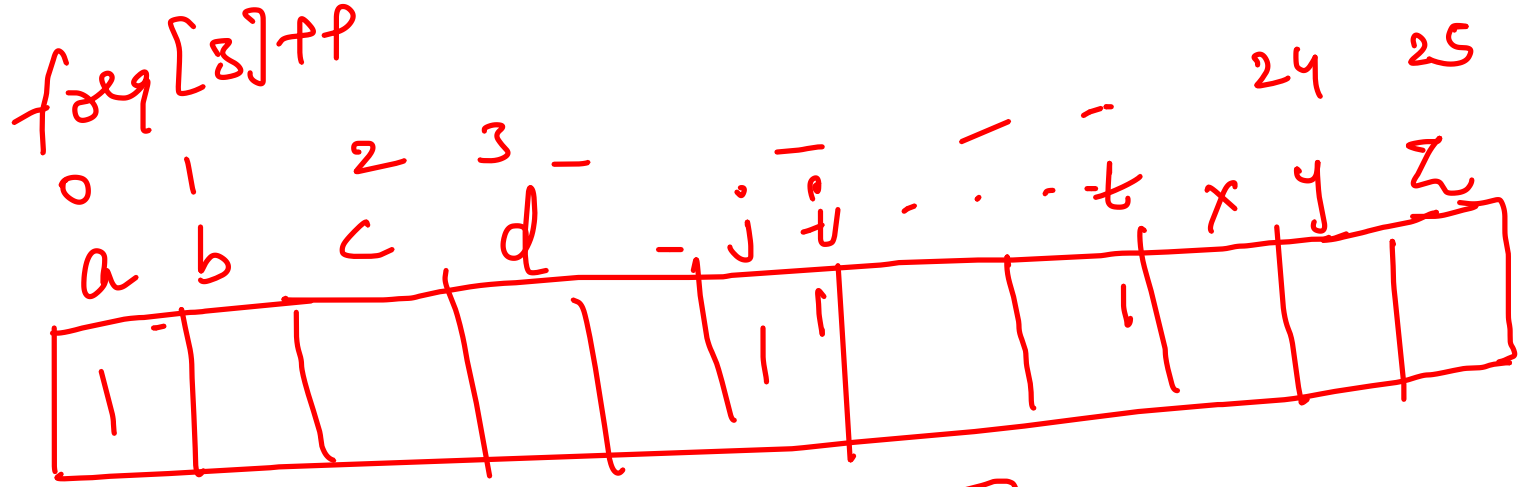
op → 1

freq =



"Jatin kumar"

freq[4]++
freq[8]++



'a' - 'a'
[97-97 = 0]

[j = 1 u = 1 v = 1
m = 1 m = 1
a = 2
t = 1 k = 1]

Minimum diff - 7.

9, 4, 1, 7

0 1 2 3

int diff = arr[i+k-1] - arr[i]

k=3.

i=0, arr[0+3-1] - arr[0]

arr[2] - arr[0]

7 - 1 = 6

i=1, arr[1+3-1] - arr[1]

arr[3] - arr[1]

9 - 4 = 5

1, 4, 7, 9

0 1 2 3

k=4.

i=0

arr[0+4-1] - arr[0]

arr[3] - arr[0]

9 - 1 = 8