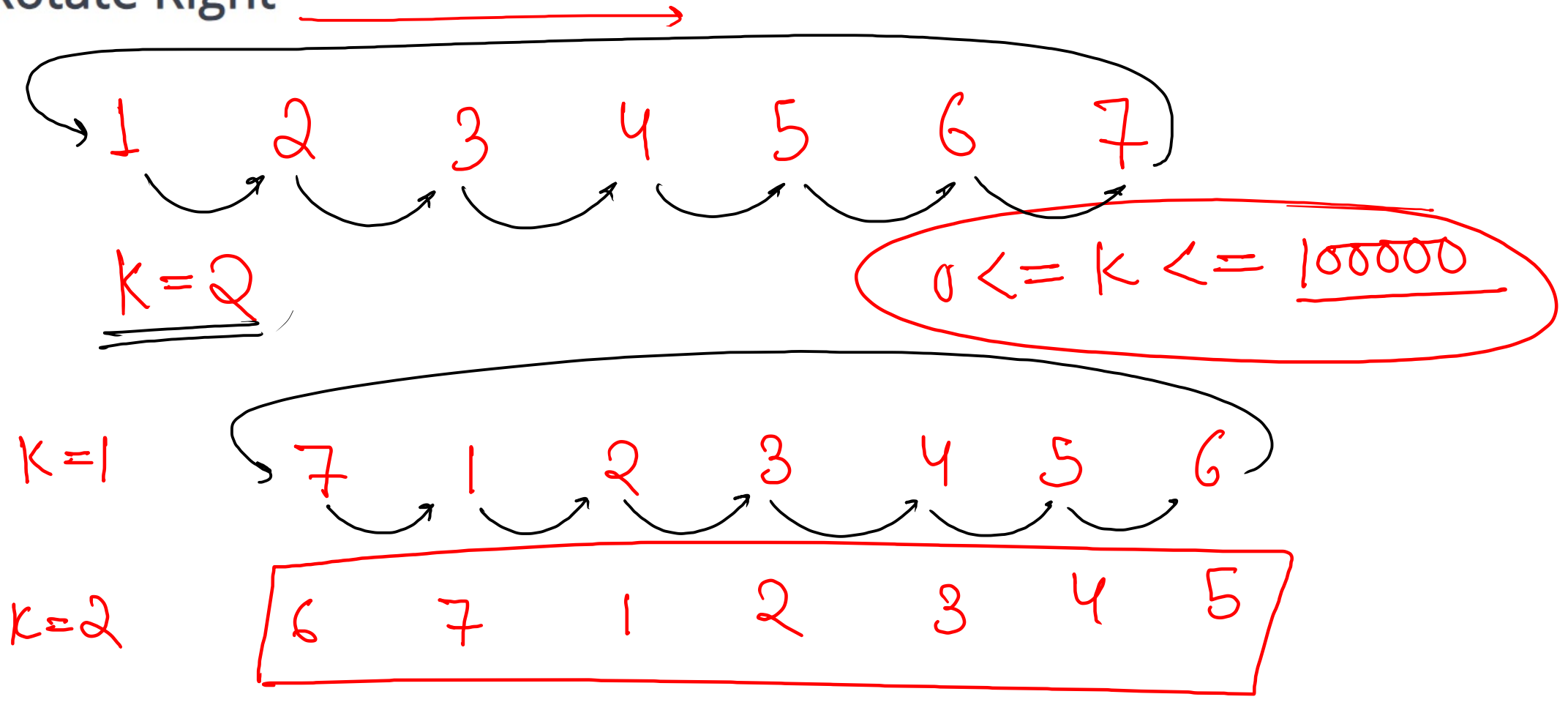
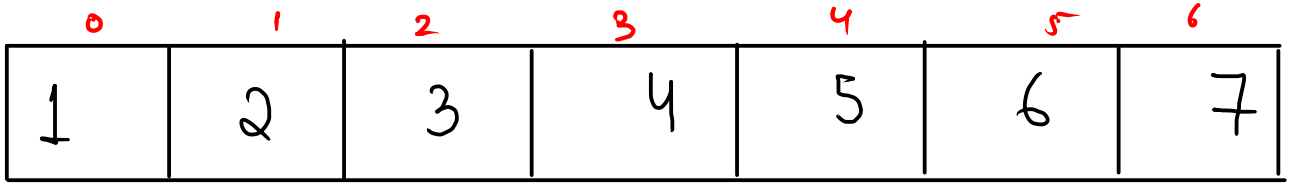
Rotate Right



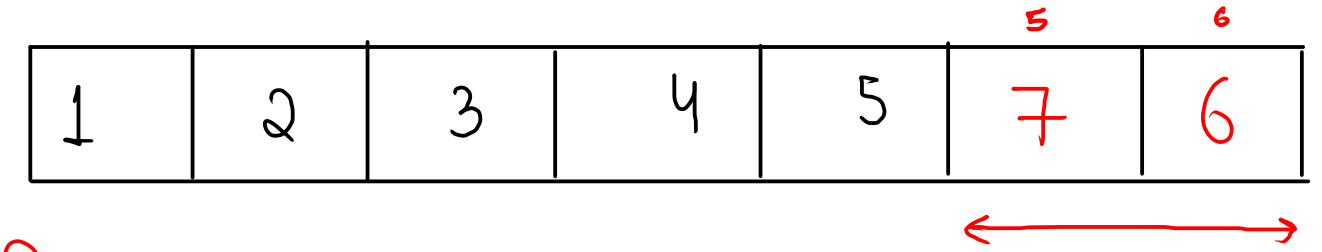
K=2

arr



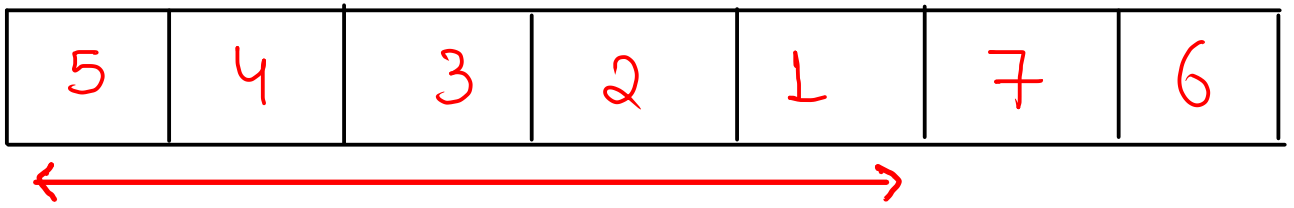
n = 7

Step 1 Reverse k elements from the last



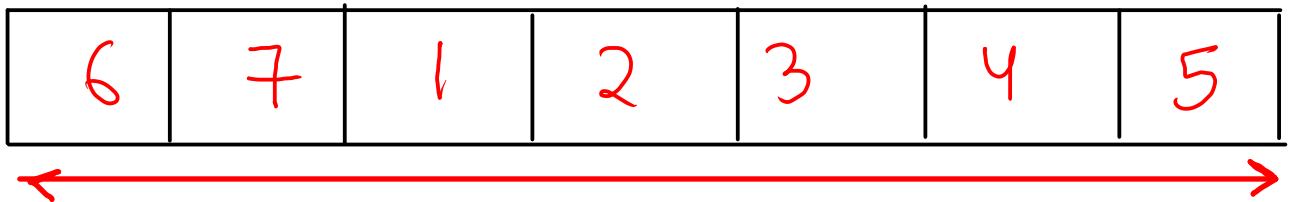
ei = n-1
si = n-K

Step 2 Reverse the remaining array



si = 0
ei = n-K-1

Step 3 Reverse the whole array



si = 0
ei = n-1

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int k = scn.nextInt();

    // main logic
    // step1
    reverse(arr, n - k, n - 1);
    // step2
    reverse(arr, 0, n - k - 1);
    // step3
    reverse(arr, 0, n - 1);

    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}

public static void reverse(int[] arr, int si, int ei) {
    while (si < ei) {
        swap(arr, si, ei);
        si++;
        ei--;
    }
}

public static void swap(int[] arr, int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}
```

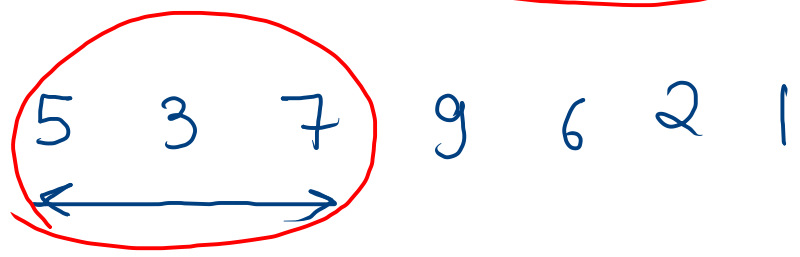
n=7
K=4

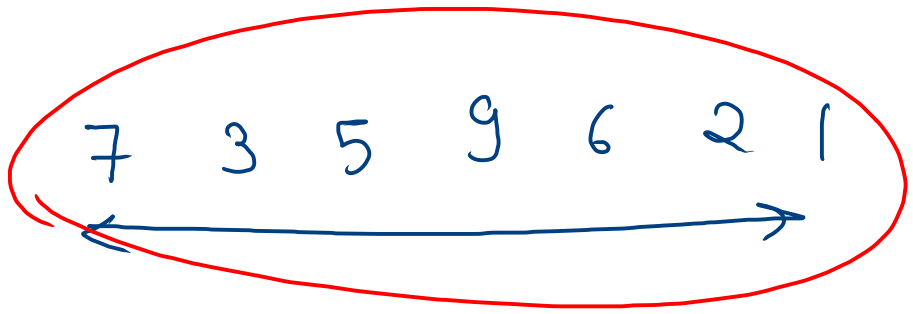
// reverse last K elements
// reverse rest of element
// reverse whole array

si	ei
3	6
0	2
0	6

arr 5 3 7 1 2 6 9

K=4





ans 1 2 6 9 5 3 7

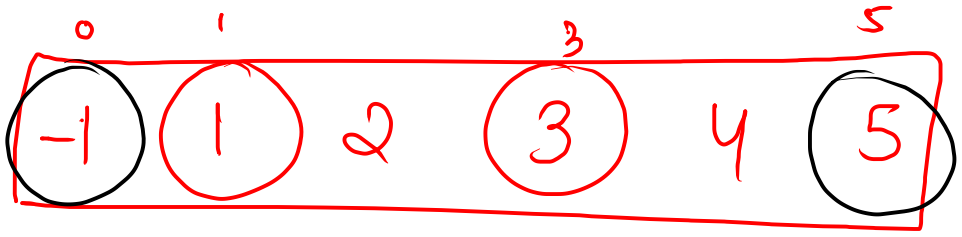
T.C = O(N), S.C = O(1)

Reach Target

n = 6

arr = -1 1 2 3 4 5 (indices 0 1 2 3 4 5)

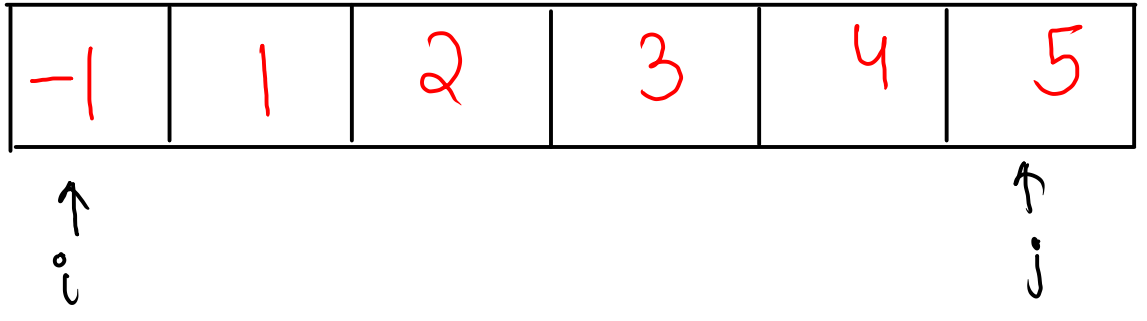
tar = 4

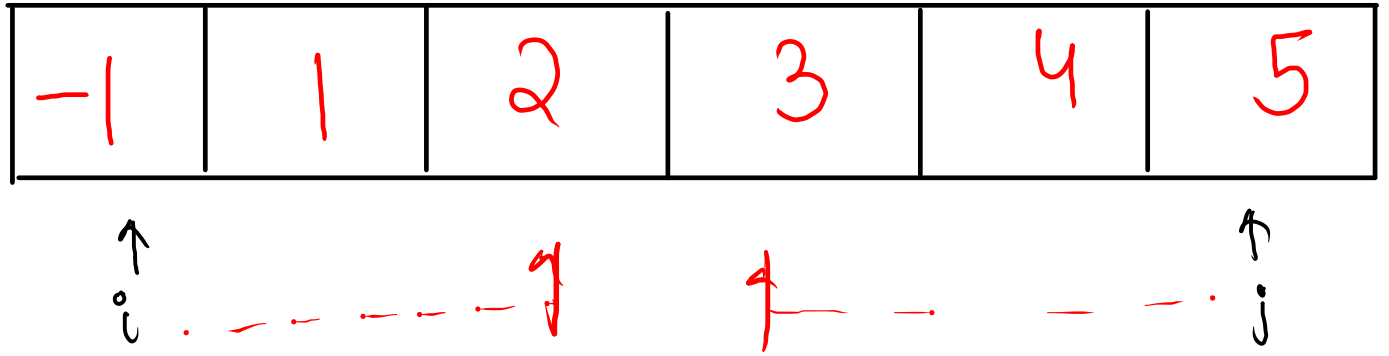


idx { 0 + 5 = 4
      1 + 3 = 4

(-1) + (5) = 4 target
 0   5

(1) + (3) = 4 target
 1   3





T.C = O(N)

S.C = O(1)

posibilities

arr[i] + arr[j] = sum

1) sum > target ; j--

2) sum < target ; i++

3) sum == target ; print ans = i and j

i++ , j--;

sum = 5 + (-1) = 4 == target

sum = 1 + 4 = 5 > target

sum = 1 + 3 = 4 == target

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int target = scn.nextInt();

    targetSum(arr, n, target);
}

public static void targetSum(int[] arr, int n, int target) {
    int i = 0;
    int j = n - 1;
    while ( i < j ) {
        int sum = arr[i] + arr[j];
        if (sum == target) {
            System.out.println(i + " " + j);
            i++;
            j--;
        } else if (sum > target) {
            j--;
        } else {
            i++;
        }
    }
}
```

Target Sum

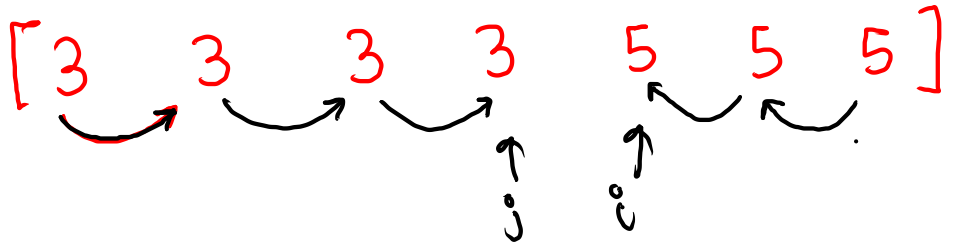
```java
public static void targetSum(int[] arr, int n, int target) {
    Arrays.sort(arr);   // sorted
    int i = 0;
    int j = n - 1;
    while ( i < j ) {
        int sum = arr[i] + arr[j];
        if (sum == target) {
            System.out.println(arr[i] + " " + arr[j]);

            // handle repetation
            while ( i < j && arr[i] == arr[i + 1] ) {
                i++;
            }
            while ( i < j && arr[j] == arr[j - 1] ) {
                j--;
            }

            i++;
            j--;
        } else if (sum > target) {
            j--;
        } else {
            i++;
        }
    }
}
```

I

II

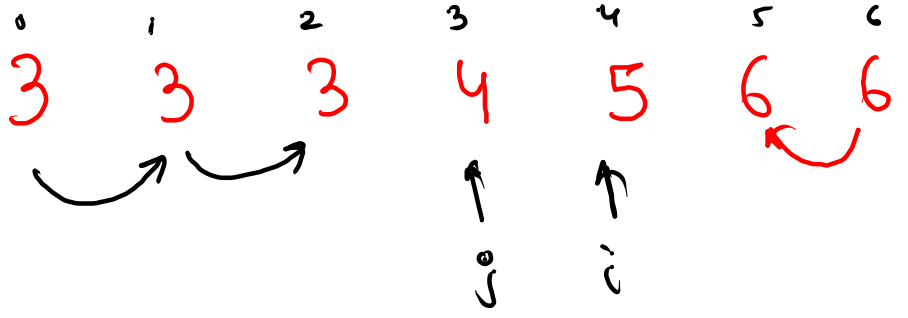
Imp

target = 8



sum = 3+5 = 8 (3, 5)

target = 9



sum = 3+6 = 9 (3, 6)

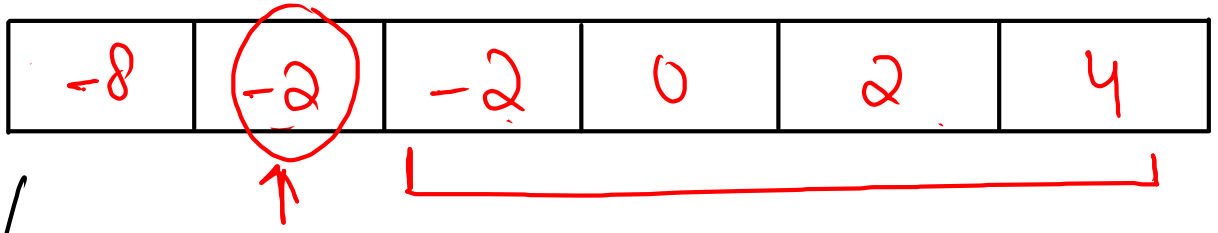
sum = 4+5 = 9 (4, 5)

$T.C = O(N + N \log N) \cong O(N \log N)$

3 Sum

sum of 3 different element

arr



-2 -2 4

-2 0 2

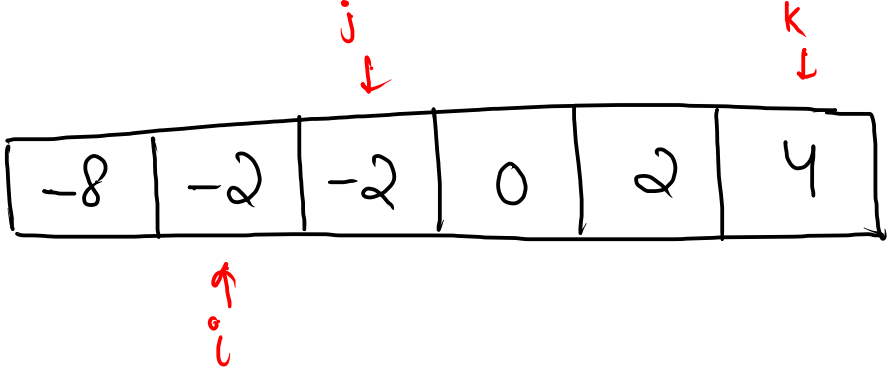
-2 0 2

$$\Rightarrow arr[i] + arr[j] + arr[k] = 0$$

$$\Rightarrow arr[i] + arr[j] = \underbrace{-arr[k]}_{\text{target}}$$

```java
public static void targetSum(int[] arr, int n) {
    Arrays.sort(arr);       // nlogn
    for (int i = 0; i < n; i++) {
        int target = -1 * arr[i];
        int j = i + 1;
        int k = n - 1;
        while ( j < k ) {
            int sum = arr[j] + arr[k];
            if (sum == target) {
                System.out.println(arr[i] + " " + arr[j] + " " + arr[k]);
                j++;
                k--;
            } else if (sum > target) {
                k--;
            } else {
                j++;
            }
        }
    }
}
```

same



i=0, target = 8 ——— (i, j, k)

i=1, target = 2 (-2, -2, 4)

(-2, 0, 2)

i=2, target=2 (-2, 0, 2)

i=3, target=0 ———

i=4, target=-2 (-2 ———)

i=5, target=4 (-4 ———)