arr = [5, 3, 7, -2, -8, 19, 10, 4, 9, 12, 15, 13]

sorted array

K = 4

n = 12

	0	1	2	3	4	5	6	7	8	9	10	11
arr =	-8	-2	3	4	5	7	9	10	12	13	15	19

start, end

Note:- when to stop :- $i <= (n-K)$

ans = ~~∞~~ ~~12~~ ~~7~~ 4

diff = ~~12~~ ~~7~~ ~~4~~ ~~5~~ ~~5~~ ~~5~~ ~~4~~ ~~5~~ 7

start = i ✓

end = i + K - 1

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 k = scn.nextInt();
    int ans = miniDiff(arr, n, k);
    System.out.println(ans);
}
public static int miniDiff(int[] arr, int n, int k) {
    Arrays.sort(arr);
    int ans = Integer.MAX_VALUE;
    for (int i = 0; i <= (n - k); i++) {
        int start = arr[i];
        int end = arr[i + k - 1];
        int diff = end - start;
        if ( diff < ans ) {
            ans = diff;
        }
    }
    return ans;
}
```

operation:- $n\log n + (n-k)$

$T.C = O(n\log(n) + n - k)$

$T.C \cong O(n\log(n))$

Form the largest number

$n = 4$

$arr = [4, 46, 8, 9]$

arr = 98464 no.

$\Rightarrow$ ["4", "46"] ✓

a = "4"

b = "46"

446
464 } ✓

str1 = a + b

str2 = b + a

return str2 - str1

arr = [4, 46, 8, 9] no.

step1 convert int array to String array

arr1 = ["4", "46", "8", "9"]

step2 sort according to requirement

```
Arrays.sort (arr1, (a, b) -> {
    String num1 = a+b;
    String num2 = b+a;
    return num2.compareTo(num1);
});
```

arr1 ["9", "8", "46", "4"]

String ans = "" + "9"
= "9" + "8"
= "98" + "46"
= "9846" + "4"
= "98464"

step3 convert String array into a no.

String ans = 98464

Note:- num = 55

1) ans = num + "";

2) ans = Integer.toString(num);

3) ans = String.valueOf(num);

```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();
    }
    String ans = largestNumber(arr, n);
    int result = Integer.parseInt(ans);
    System.out.println(result);
}
public static String largestNumber(int[] arr, int n) {
    String[] arr1 = new String[n];
    for (int i = 0; i < n; i++) {
        arr1[i] = Integer.toString(arr[i]);
    }

    Arrays.sort( arr1, (a, b) -> {
        String num1 = a + b;
        String num2 = b + a;
        return num2.compareTo(num1); // num2 - num1
    } );

    String ans = "";
    for (int i = 0; i < n; i++) {
        ans = ans + arr1[i];
    }

    return ans;
}
```

T.C = O(nlog(n))

["46","4"]

a = "46"

b = "4"

num1 = "464"

num2 = "446"

num2 - num1

-ve order