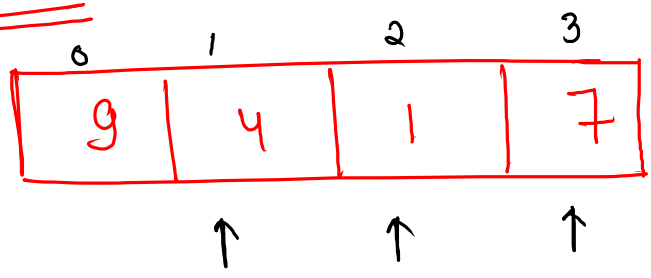
Minimum difference 7

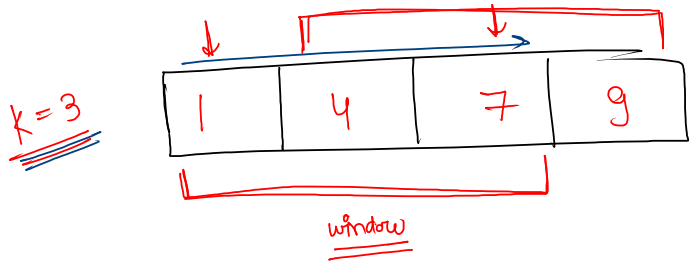
take some combinations of K values
↳ each, (largest − smallest)
min

n = 4



K = 3

- 9, 4, 1 = 8
- 9, 4, 7 = 5 ✓✓
- 9, 1, 7 = 8
- 4, 1, 7 = 6

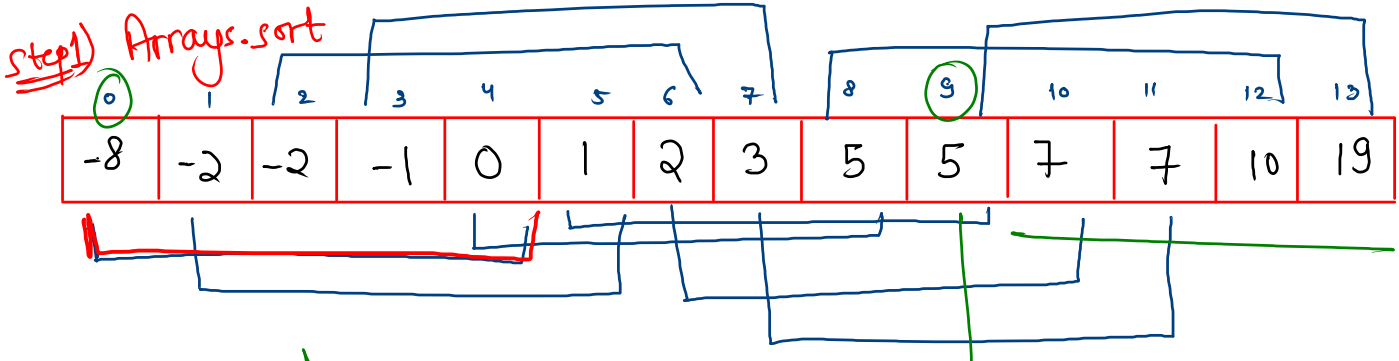


ans = ~~6~~ 5

---

arr , k=5, n=14

5	3	7	-2	-8	19	10	0	1	2	5	-2	-1	7



ans = ~~0~~ ~~8~~ 3

<= n-k

$k = 5$

pairs = $i = 0$

$0, 4 \longrightarrow 0 + 5 - 1$

$1, 5 \longrightarrow 1 + 5 - 1$

$2, 6 \longrightarrow 2 + 5 - 1$

$3, 7 \longrightarrow 3 + 5 - 1$

$4, 8$

$\vdots$

$i + k - 1$

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();
    System.out.println(miniDiff(arr, n, k));
}

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 mini = arr[i];
        int maxi = arr[i + k - 1];
        int diff = maxi - mini;
        if ( diff < ans ) {
            ans = diff;
        }
    }
    return ans;
}
```

T.C

O(NlogN)

Form the largest number

n = 4

4 46 8 9

98644

ans = 98464