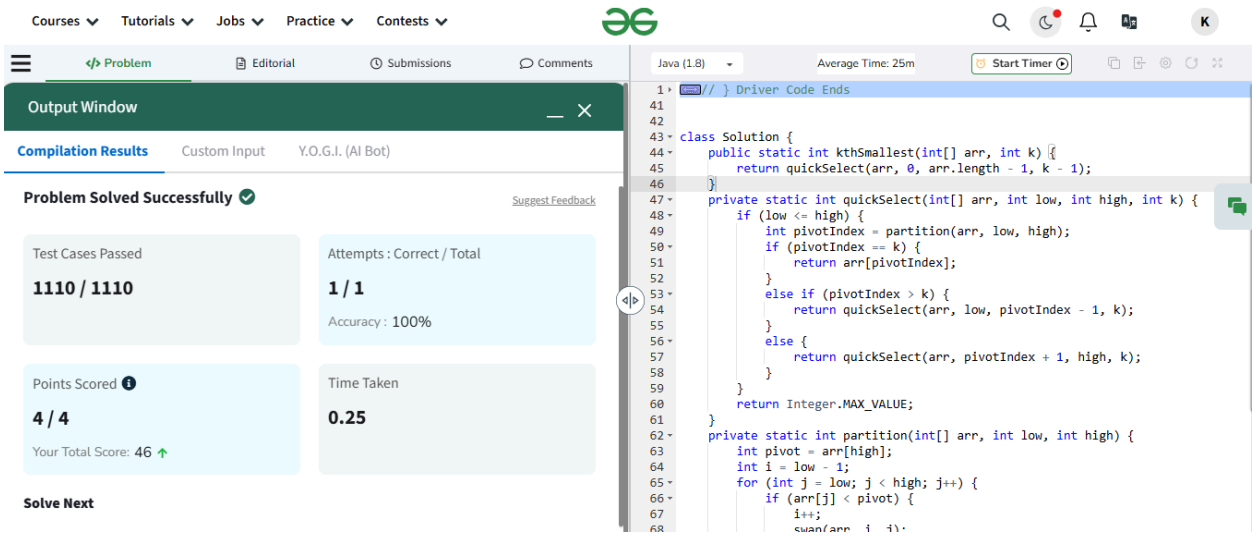
DSA Coding Practice-4

1.Kth Smallest Element:

```
class Solution {
    public static int kthSmallest(int[] arr, int k) {
        return quickSelect(arr, 0, arr.length - 1, k - 1);
    }
    private static int quickSelect(int[] arr, int low, int high, int k) {
        if (low <= high) {
            int pivotIndex = partition(arr, low, high);
            if (pivotIndex == k) {
                return arr[pivotIndex];
            }
            else if (pivotIndex > k) {
                return quickSelect(arr, low, pivotIndex - 1, k);
            }
            else {
                return quickSelect(arr, pivotIndex + 1, high, k);
            }
        }
        return Integer.MAX_VALUE;
    }
    private static int partition(int[] arr, int low, int high) {
        int pivot = arr[high];
        int i = low - 1;
        for (int j = low; j < high; j++) {
            if (arr[j] < pivot) {
                i++;
                swap(arr, i, j);
            }
        }
        swap(arr, i + 1, high);
        return i + 1;
    }
    private static void swap(int[] arr, int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}
```



2.Minimize the Heights II: