

Problem 3: Find K-th largest element in a stream

```
import heapq
```

```
class TreeNode:
```

```
    def __init__(self, val):
```

```
        self.val = val
```

```
        self.left = None
```

```
        self.right = None
```

```
class KthLargest:
```

```
    def __init__(self, k, nums):
```

```
        self.k = k
```

```
        self.heap = []
```

```
        for num in nums:
```

```
            self.add(num)
```

```
    def add(self, val):
```

```
        if len(self.heap) < self.k:
```

```
            heapq.heappush(self.heap, val)
```

```
        else:
```

```
            heapq.heappushpop(self.heap, val)
```

```
    def find_kth_largest(self):
```

```
        return self.heap[0]
```

```
# Test case
```

```
nums = [4, 5, 8, 2]
```

```
k = 3
```

```
kth_largest = KthLargest(k, nums)
```

```
print(kth_largest.find_kth_largest())
```

