K-th Largest Element in Array

We want the k-th largest element in an unsorted array without sorting the entire array.

Approaches:

1. Min-Heap (size k):
   - Keep a min-heap of size k.
   - Traverse nums:
     - Push current element into the heap.
     - If heap size exceeds k, pop the smallest.
   - At the end, the heap's root is the k-th largest.

   Time: $O(n \log k)$

   Space: $O(k)$
2. Quickselect (Hoare's selection algorithm):
   - Partition the array like quicksort around a pivot.
   - After partition, check pivot's index:
     - If it is $n-k$, that element is the answer (since we want k-th largest).
     - Else recurse into the half that contains $n-k$.
   - Expected time: $O(n)$
   - Worst case: $O(n^2)$, but random pivoting avoids this.

Both solutions avoid full sorting ($O(n \log n)$).

- Use heap if you want guaranteed $O(n \log k)$.
- Use quickselect if you want average $O(n)$ and minimal extra memory.