973. K Closest Points to Origin

```
from heapq import *
class Solution:
    def kClosest(self, points: List[List[int]], K: int) -> List[List[int]]:
        result = []
        for x, y in points:
            if len(result) == K:
                heappushpop(result, (-x**2 - y ** 2, x, y))
            else:
                heappush(result, ((-x**2 - y ** 2, x, y)))
        return map(lambda a: [a[1], a[2]], result)
```

215. Kth Largest Element in an Array

```
from heapq import *
class Solution:
    def findKthLargest(self, nums: List[int], k: int) -> int:
        result = []
        for num in nums:
            if len(result) == k:
                heappushpop(result, num)
            else:
                heappush(result, num)
        return result[0]
```

215.1 quick select

```
class Solution:
    def findKthLargest(self, nums: List[int], k: int) -> int:
        l, r = 0, len(nums) - 1
        while l <= r:
            mid = self.partition(l, r, nums)
            if mid == k - 1:
                return nums[mid]
            elif mid > k - 1:
                r = mid - 1
            else:
                l = mid + 1




    def partition(self, l, r, nums):
        if l == r:
            return l
        temp = l
        l += 1
        while l < r:
            while nums[l] > nums[temp] and l < r:
                l += 1
            while nums[r] <= nums[temp] and l < r:
                r -= 1
            if l != r:
                nums[r], nums[l] = nums[l], nums[r]
        if nums[l] > nums[temp]:
            nums[l], nums[temp] = nums[temp], nums[l]
            return l
        else:
            nums[l - 1], nums[temp] = nums[temp], nums[l - 1]
            return l - 1
```

414. Third Maximum Number

```
from heapq import *
class Solution:
    def thirdMax(self, nums: List[int]) -> int:
        result = []
        for num in nums:
            if num not in result:
                if len(result) == 3:
                    heappushpop(result, num)
                else:
                    heappush(result, num)
        return max(result) if len(result) < 3 else result[0]
```

658. Find K Closest Elements

```
from heapq import *
class Solution:
    def findClosestElements(self, arr: List[int], k: int, x: int) -> List[int]:
        result = []
        for num in arr:
            if len(result) == k:
                heappushpop(result, (-abs(num - x), -num, num))
            else:
                heappush(result, (-abs(num - x), -num, num))
        return sorted(list(map(lambda a: a[2], result)))
```