Smart Coding & Interview Series
Top-20 Basic Program
(Heap & Priority Queue Applications)

First, understand the solution building strategies and coding for the problems in LIVE/VIDEO session and then you apply those strategies discussed in LIVE/VIDEO session to solve the following problems. Use your favourite language(C/C++/Java/C#/Python/Scala) for coding.

1) Top-k Smallest Elements in Realtime DataStream: Find lowest top k frequent number s in realtime data stream. Implement two methods for Class:

1. add(number). Add a new number in the data structure.
2. topk(). Get the current top *k* smallest elements.

2) Kth Smallest Sum in Two Sorted Arrays: Given two integer arrays sorted in ascending order and an integer k. Define sum = a + b, where a is an element from the first array and b is an element from the second one. Find the kth smallest sum out of all possible sums.
Example
Given [1, 7, 11] and [2, 4, 6].
For k = 3, return 7.
For k = 4, return 9.
For k = 8, return 15.

3) Merge k Sorted Lists: Find an efficient algorithm to merge *k* sorted linked lists and return it as one sorted list.
Example:
Input:
[
  1->4->5,
  1->3->4,
  2->6
]
Output: 1->1->2->3->4->4->5->6
Source : *https://leetcode.com/problems/merge-k-sorted-lists/*

4) Kth Smallest Element in a Sorted Matrix: Given a *n* x *n* matrix where each of the rows and columns are sorted in ascending order, find the kth smallest element in the matrix. Note that it is the kth smallest element in the sorted order, not the kth distinct element.
Example:
Matrix=[
[01, 05, 09]
[10, 11. 13]
[12, 13, 15]

]
K=8
Return 13.
Source:[*https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/description/*](https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/description/)

5) Sliding Window Maximum: Given an array nums, there is a sliding window of size *k* which is moving from the very left of the array to the very right. You can only see the *k* numbers in the window. Each time the sliding window moves right by one position. Return the max sliding window.
Example:
Input: nums = [1,3,-1,-3,5,3,6,7], and k=3
Output: [3,3,5,5,6,7]
Source : [*https://leetcode.com/problems/sliding-window-maximum/description/*](https://leetcode.com/problems/sliding-window-maximum/description/)