Problem Solving Workshop #25 Tech Interviews and Competitive Programming Meetup
June 18, 2017 https://www.meetup.com/tech-interviews-and-competitive-programming/
Instructor: Eugene Yarovoi (can be contacted through the group Meetup page above under Organizers)

More practice questions: leetcode.com, glassdoor.com, geeksforgeeks.org
Books: Elements of Programming Interviews, Cracking the Coding Interview
Have questions you want answered? Contact the instructor, or ask on Quora. You can post questions and follow the instructor and other people who write about algorithms.

Try to find optimized solutions, and provide a time and space complexity analysis with every solution.

Problem #1, "Bitwise OR of a Range"

(If you're unfamiliar with bitwise operators, search for "bitwise operators" before starting this problem. You only need to know the basics. Make sure you know what "bitwise OR" means. Bitwise operators are easy to understand, and you might be able to get the concept in 10 minutes.)

Given two 64-bit integers A and B, with B > A, give an algorithm to compute A | (A+1) | (A+2) | ... | (B-1) | B, where | is the bitwise OR operator. In other words, we are computing the bitwise OR of all the integers between A and B, inclusive of A and B. Consider that A and B could be large values that are far apart. Naive solution is O(B-A), but a good time complexity is O(log B), though it is possible to solve even faster.

Example Input: A = 3, B = 6
Output: 7
Explanation: Using binary notation to express the numbers, we are performing 011 | 100 | 101 | 110 = 111, which is 7 in decimal.

Problem #2, "Telecom Towers"

N telecom towers are arranged in a line, along a road, and numbered 0 through N-1 contiguously. Any car that is on the road will, at any given position, be receiving the transmissions from some contiguous subrange of towers. For example, there may be one spot on the road where a car would receive towers 2-5, and another spot where a car would receive towers 1-7. But it cannot be, for example, that towers 8, 9, and 11 are received in the same spot, but 10 is not. You don't know which towers will be in range where -- at any point, you have to assume that the towers in range could be any contiguous subrange of towers.

For each tower, you are to choose one frequency at which the tower will broadcast. To enable interference-free transmission, at every point in the road (i.e. for any possible contiguous subrange of towers), there must exist a frequency that is unique among all the frequencies in range at that point. This could be achieved by having every tower use a different frequency, but if we want to re-use frequencies as much as possible, given N, what is the optimal assignment of frequencies to towers?

Example Input: N = 3
Output: [1, 2, 1]
Explanation: 1 and 2 represent distinct frequencies in the output. [1, 2, 1] is a better assignment than [1, 2, 3] because it uses fewer frequencies (2 instead of 3). [1, 2, 1] is valid because the contiguous subranges of that array are [1], [2], [1], [1, 2], [2, 1], and [1, 2, 1], and every single one of these contiguous subranges has at least one frequency that is unique in that subrange (1, 2, 1, 1 or 2, 1 or 2, and 2, respectively).

Example Input: N = 5
Output: [1, 2, 3, 1, 3] (One of many possible solutions involving 3 frequencies. Impossible to solve with just 2. You can verify the solution by checking that in every contiguous subrange, some frequency is unique.)

Guidance: if you're struggling with the problem, try solving by hand for some small examples, and look for patterns.

Problem #3, "Scattered Points"

K spheres of radius R (common radius among all spheres) are placed in 3-dimensional space, such that none of the spheres overlap. Then, 1 point is chosen from inside each sphere (in an arbitrary fashion). You are then given the chosen points (in no particular order), the centers of the spheres (in no particular order), and the value of R. Your goal is to output a mapping showing to which sphere each point corresponds.

(That the spheres don't overlap, and that the points are chosen from inside the spheres, implies that any point can only correspond to one sphere.)

Example Input: points = [(2.8, 2.6, 3.2), (0, 0, 0.4)], spheres = [(0, 0, 0), (3, 3, 3)] , R = 1
Output: {point 0 -> sphere 1, point 1 -> sphere 0}
Explanation: Point 0 corresponds to sphere 1 because (2.8, 2.6, 3.2) is inside the sphere with center (3, 3, 3) and radius 1. Point 1 corresponds to sphere 0 because (0, 0, 0.4) is inside the sphere with center (0, 0, 0) and radius 1.