1. **Top 100:** Given an array and $k$, find its $k$ largest elements. What if the array is too big to fit in memory?

2. **String: Anagram?** Are two strings permutations of each other?

3. **Find the missing numbers:** You are given a sequence of $n$ distinct integers, in sorted order, starting at 1 and ending with $n + k$. Find the $k$ missing numbers in the sequence. Your algorithm should be faster than linear time if $k$ is small.

4. **Partition into subsequences:** Given a sorted sequence (with duplicates) of integers, can it be partitioned into subsequences of consecutive integers, where each subsequence has at least 3 integers? For example, if $A = \{1, 2, 3, 3, 4, 4, 5, 5\}$, then it can be partitioned into $A_1 = \{1, 2, 3, 4, 5\}$ and $A_2 = \{3, 4, 5\}$. But $B = \{1, 2, 3, 4, 4, 5\}$ cannot be partitioned as required.

5. **DP: Monotonic chain:** Given a matrix of numbers, find a chain of monotonically increasing numbers. Matrix elements can be chained together with 4 adjacent elements (up, down, left, right). Problem can also be formulated with 8 neighbors of a node, or with obstructions.

   4 neighbors of X: 8 neighbors of X: Obstructions(\ ):

   * * * * | * *
   * X * * X * * X |
   * * * * * * * |

6. **Lines and squares:** Given a list of horizontal or vertical line segments in the plane, find the number of squares formed.
7. **String: Breaking into words:** Given a set of words (dictionary), and a set of strings, write a function that decides if each string can be broken into substrings that are words in the dictionary. What about the problem of counting the number of solutions?

8. **Pirates of the Caribbean:** You are travelling in an archipelago on a ship, whose hull has thickness $H$. The islands, and the sea routes among them, is modeled as a graph, $G = (V, E)$. Each edge $e \in E$ takes time of $t(e)$ to travel, and wears down the hull of the ship by $w(e)$. Given nodes $s$ and $t$, find if it is possible to go from $s$ to $t$ such that the ship survives the trip, i.e., the sum of the amount by which the edges wear the ship’s hull is less than $H$. Find the shortest time for such a trip.

9. **Recursion practice:** Convert a doubly-linked, circular, sorted list into a balanced binary search tree (prev=left, next=right) in $O(n)$ time, without allocating extra space. Reverse problem: BST $\rightarrow$ sorted list.

10. **String: Tokenizer:** Given a set of keywords (say, of a programming language), write a function that takes a string as input and finds if it is a keyword or an identifier (such as name of variable, class, or function). This function will be called many times, and the keywords are fixed. What is a good implementation of the function?

11. **Egg drop problem:** if an egg is dropped from a floor of the building, it either breaks or survives the fall. If an egg survives, then it would have survived any lesser fall, and the egg can be used for another experiment. If the egg breaks, then any greater fall would have broken it as well, and any subsequent trials needs new eggs. The problem is to find the maximum floor from which an egg survives a fall. Given $f$ floors, and $n$ eggs, what is the fewest number of trials needed to find the answer? Alternately, given $n$ eggs and $t$ trials, what is the maximum number of floors that can be explored?

13. **Submatrix with maximum sum:** Given an array $A[M \times N]$ of numbers, find a subarray whose sum is a maximum among all subarrays of $A$.

14. **External: substream with $k$ different characters:** Given a (really long) stream of characters, find the length of a longest substream that has at most $k$ different characters?

15. **Graph: Number of shortest paths:** Find number of shortest paths from source to destination in a graph or an $M \times N$ grid (maybe with obstructions).

16. **DP: Treasure hunt on a grid:** Given a grid of numbers, find a path from NW corner to SE corner, moving only right or down, such that the sum of the numbers on the path is a maximum. Related problem: How many paths are there with a given sum $K$?

17. **String: Extending to a palindrome:** Shortest string that can be added as a prefix (or suffix) to a given string to make it a palindrome.

18. **Max sum:** Given an array of numbers, find the maximum sum of a subsequence with the constraint that no two numbers in the subsequence are adjacent in the array.

19. **Searching in sorted arrays:** Given a sorted array $A[1..n]$, output its distinct elements $D[1..k]$ and their counts $C[1..k]$. For example, if $A = \{1, 4, 4, 5, 5, 5, 8, 8, 9, 9\}$, then $D = \{1, 4, 5, 8, 9\}$ and $C = \{1, 2, 4, 2, 2\}$. Related problem: Find the $i$th smallest distinct element of a sorted array. In the above example, if $i = 3$, the output is 5.
20. **Word maze:** Given a collection of words, find which of the words exist in a given matrix of letters, where each word is formed by adjacent letters on a line (horizontal, vertical, or diagonal).

21. **Bitonic arrays:** Given a bitonic array, find the point of inflection. An array is bitonic if it is made of a monotonically increasing subarray that is followed by a decreasing subarray.

22. **Number theory: perfect powers:** Determine whether a number is a nontrivial power, \( n = p^q, (q > 1) \). More generally, write a program to generate all perfect powers, in increasing order, between 1 and \( 10^9 \).

23. **DP: Subset sum, Knapsack, Set partition, Balanced set partition:** **Subset sum:** Given a set \( S \) of positive numbers and a target \( t \), is there a subset of \( S \) whose sum is \( t \)?

   **Knapsack:** Given \( S \) and \( t \), find a subset whose sum is largest, but no bigger than \( t \).

   **Set partition:** Given \( S \), can it be partitioned into two subsets which have equal sum?

   **Balanced set partition:** Given \( S \), can it be partitioned into two subsets of equal cardinality and equal sum?

24. **Strings, substrings and subsequences:** (a) Search for a given set of substrings in a string. (b) Shortest substring of \( S \) that contains \( T \) as a subsequence. Assume that \( T \) is a subsequence of \( S \). (c) Find the number of distinct substrings of a given string. (d) Longest repeating prefix: Find longest prefix \( p \) of a string which has \( pp \) as a prefix.

25. **Number of different BST’s:** Given \( n \), how many structurally different BST’s (binary search trees) are there, that store values 1..\( n \)?

26. **Maximum rectangle in a histogram:** Given a histogram (vertical bar chart), as an array of integers, find the maximum area rectangle contained in the bar chart.