GCS: Problem solving session

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(\|):

\[
\begin{array}{cccc}
  * & * & * & * \\
  * & X & * & * \\
  * & * & * & * \\
  * & * & * & * \\
\end{array}
\]

String: Anagram?: Are two strings permutations of each other?

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?

String: Dynamic dictionaries: Operations on a dictionary: add(word), remove(word), contains(word). Implementation?

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 is a good implementation of the function?

String: Set operations: Given two linked lists of strings, output a list of strings that are in both lists (intersection of two unsorted lists).

External problem: distinct elements: Given a large file (orders of magnitude bigger than available memory), where each line has one or more integers, find the lines of the file that are distinct. Two lines are considered to be the same if they contain the same integers (possibly in different order). For example, ”1 2 3 3” is the same as ”3 1 3 2”. Design an algorithm that tries to minimize the number of passes over the file.

Does a linked list end in a cycle?: Fun problem: Suppose you are given a singly linked list, where by a programmer error, the last node of the list may have its next pointer set to an element of the list. The problem is to find if a given list ends in a cycle or not, in time \(O(n)\) where the list has \(n\) nodes, using only \(O(1)\) extra space. The standard solution to this problem uses 2 iterators that run at different speeds (tortoise and hare solution). Can this problem be solved with only one iterator?

String: distinct elements: Are all characters in a character array (String in C) distinct? Solve using only \(O(1)\) extra space. Same problem: Given an array of integers, are all its elements distinct?
String: \(\{a, b, c\}^*\): Given a string as input, find a longest substring that is composed of at most 3 different characters. In general, solve the problem for “at most \(k\) different characters.”

Recursion: sum of digits: Given \(n > 0\), find sum of digits of \(n \mod 10\).

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.

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 \(10^9\).

Subarray with maximum sum: Given an array \(A[1..n]\), find a subarray \(A[i..j]\) with maximum sum (over all subarrays of \(A\)).

Subsequence with maximum sum: Given an array \(A[1..n]\), find a subsequence of \(A\) with maximum sum (over all subsequences of \(A\)).

Subsequence with maximum sum: What if adjacent elements of the output subsequence must have a difference of more than \(k\) (say, \(k = 5\)). What if the subsequence must be monotonic?

DP: 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 \(A\).

External: substream with \(k\) different characters: Given a (really long) stream of characters, find a shortest substring that has at least \(k\) different characters? Length of a longest substream that has at most \(k\) different characters? Given a 0/1 array, find longest contiguous 1s that can be obtained after flipping up to \(k\) 0s into 1s.

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?

String: longest/shortest substring with property \(P\): Longest substring all of whose characters are distinct.
Shortest substring that has the entire alphabet.
Smallest subarray of an array that contains all its elements.

External: \(k\) largest elements: Given a stream whose length is much bigger than memory, find its \(k\) largest elements (say, \(k = 100\)). Assume that \(k\) is smaller than available memory.
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).

Palindromes, subsequences: Longest palindromic subsequence of a sequence. Longest subsequence that forms a balanced set of parentheses. Breaking a string into smallest number of palindromes.

DP: Frog crossing river: Fun problem: A frog wants to cross a pond of width $W$. The frog can jump a maximum distance of $D$ in one jump. Luckily there are leaves falling from a tree on to the surface of the pond and the frog can jump on to and from the leaves. A leaf falls every second, and you are given an array of locations where the leaves are falling. $A[i] = k$ means, at second $i$, a leaf lands at a distance of $k$ from the bank on which the frog is initially located. Find the earliest time at which the frog is able to cross the pond. Variations: Leaves stay afloat only for limited time before its surface gets wet and unsuitable to use as landing pad.

Rotational symmetry: Simple/Fun: Does a number have rotational symmetry (88,69,000,16191)? Rotate the number 180 degrees about its centre and you get the same number.

Trees: Level order traversal of a tree: Given a binary tree as input, return a list of lists (one list for each level).

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 $K$?

Counting triangles: No easy answers: Count number of triangles in a graph.

2-player coin game: Given a row of coins, with values $A[1..n]$, two players play the following game. The players take alternate turns. At each turn, a player can choose to take a coin at the left end, or a coin at the right end, or both coins at the two ends. The game is played until all coins are taken. What strategy should a player employ if she wants to maximize the total value of the coins she gets?

Mnemonics for phone numbers: Given a dictionary of words and a phone number, check if the given phone number maps to a combination of words from the dictionary. See your phone for a mapping of each number to letters of the alphabet.

Tree traversal: Reconstruct binary tree from its inorder, and postorder traversals. What if the elements are not distinct?

Group interval scheduling: Given groups of intervals, $G_1, \ldots, G_k$, where each group has 2 intervals, check if there is a collection of $k$ mutually disjoint intervals that consists of one interval from each group.

Off-line stock market: Given an initial budget, and an array of stock prices over $n$
days, find buy/sell transactions to maximize net worth on day $n$.

**String: number of distinct substrings:** Find the number of distinct substrings of a given string.

Longest repeating prefix: Find longest prefix $p$ of a string which has $pp$ as a prefix.

**Heuristics: Friend recommendation:** Friend recommendation in Social networks.

**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.

**Enumeration: finding all paths:** Find all paths in a 8x8 grid from NW corner to SE corner (with obstructions). Subsets of a set. Min-weight matchings in small graphs.

**Closest elements:** Pairs of numbers with minimum difference in unsorted array: constraints on time, space.

**Counting Sort:** Order objects by color: red, white, and blue.