**Problem solving session: Spring 2019**

**Topics covered in interviews**

Other than topics covered in Algorithms (6363) and Data Structures (5343), the following topics are common in interviews:

String algorithms: Knuth-Morris-Pratt (KMP) string matching, Boyer-Moore string matching, Manacher's algorithm for longest palindromic substring.

String data structures: Tries, Suffix trees (compressed tries).

Bit manipulations.

Finite State Machines (FSM), Regular expressions.

Programs with complex nested if statements can sometimes be simplified to a switch statement based on states of a FSM, that is easy to code and maintain.
Ex: Rearrange $1 \rightarrow 2 \rightarrow 3 \rightarrow ...$ to $1 \rightarrow 4 \rightarrow 7 \rightarrow ...2 \rightarrow 5 \rightarrow 8...3 \rightarrow 6 \rightarrow 9...$

Number theory: primes, factorization, GCD, mod arithmetic.

Enumeration: permutations, combinations with precedence constraints (e.g., number of topological orders of a DAG), exploring states of a game.

**How to tackle new problems, avoiding traps**

Try modeling given problem so a standard algorithm can be applied.

Use data structures from standard libraries, when possible.

When in doubt, avoid greedy algorithms. Designing algorithms without knowing their proof of correctness usually leads to incorrect algorithms.

Avoid using global variables to store state information in dynamic programs. Functions with arrays or lists as parameters (that are being modified during the algorithm) will lead to exponential time dynamic programs.

**Dynamic programming strategies**

Find a solution using divide and conquer. Identify all subproblems that arise when problem is solved. Design storage to store these solutions. Generate problems in order of increasing sizes (i.e., the induction order), compute their solutions without recursion, and store them.

Example: In how many ways can change of $T$ be issued with an unlimited supply of coins in the denominations $c_1..c_n$?

What if there is a limited supply of these coins, $s_1..s_n$?

Enumerate all states and calculate interaction between them. Find the best solution. Examples: Tic-tac-toe (19683), 8-puzzle (362880), 15-puzzle ($2 * 10^{12}$), Rubik's cube ($43 * 10^{18}$), Chess ($10^{45}$).

Bad ideas: greedy strategies, global variables

Algorithms designed without consideration of correctness, tend to be wrong.

Harder the problem, easier it is to design a wrong algorithm!

## Counting, Recursion

How many $n$-digit numbers are there without using the digit 5?

How many distinct binary search trees of $n$ nodes are there?

How many distinct balanced strings can be formed using $n$ pairs of parentheses?

Given an array $A[1..n]$ and a target sum $T$, how many subsets does $A$ have, with sum $T$?

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

How many times does $T$ occur as a subsequence of $S$?
Ex: S = accbcacbc, T = abc. Answer: 4.
accbcacbc, accbcacbc, accbcacbc, accbcacbc

Find the $k$th smallest of two sorted arrays in $O(\log(k))$ time.

## Hidden graph problems

Given a 2d grid map of '1's (land) and '0's (water) count the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

Longest monotonic chain on a grid of numbers.

A company wants to interview $n$ candidates, at one of $k$ offices, with a maximum of $n_i$ candidates at office $i, i = 1..k$. Given an $n \times k$ matrix, with the cost of travel for each candidate to each of the offices, find a minimum-cost assignment of the candidates to the offices.

Given a set of strings where no string is a substring of another, find a shortest string that contains all of them as substrings.
Input : [ catgc ctaagt gcta ttca atgcatc]. Answer: gctaagttcatgcatc

Given $n$ box types, with dimensions $x_i \times y_i \times z_i, i = 1..n$, find the maximum height that can be reached by stacking boxes, where a box can be placed on another box only if the face (rectangle) of the box abov,e fits within the face of the box below. Boxes can be rotated and reused.

## Data structures

Given an unsorted array, find a longest streak of consecutive integers in it.

Merge $k$ sorted lists into one sorted list.

Given an array $A[1..n]$, for each $i = 1..n$, find $bigger(i)$, the number of elements of $A[i+1..n]$, that are bigger than $A[i]$.

Regular expression matching.

Given a singly linked list, reverse its nodes, $k$ at a time.

Given $n$ processes and their parent processes (PID and PPID), write a method descendant($p$), that returns the list of processes that have $p$ as an ancestor.

## Trick questions

Given an unsorted array of positive integers, test if they are distinct integers forming a streak, using only $O(1)$ extra space.

Given a 3*3 matrix s of integers with numbers 1..9, convert it to a magic square, at minimal cost. The cost of replacing $a$ by $b$ is $|a - b|$.

Given a node in the middle of a singly linked list, delete it.

**String problems**

Data structure for storing a semi-dynamic dictionary, with operations $add(w)$ and $contains(w)$, or a dynamic dictionary with $remove(w)$.

Mnemonics for phone numbers.

Given a string $S$ and integer $k$, find a longest substring of $S$, that is composed of at most $k$ different characters.

Find the number of distinct substrings of a given string.

Find longest prefix $p$ of a string that has $pp$ as a prefix.

Shortest string that can be added as a prefix (or suffix) to a given string to make it a palindrome.

Shortest substring of $S$ that contains all characters that occur in $T$.

**Dynamic programming**

Given a set of words (dictionary), break a given string into fewest number of atoms. An atom is either a character or any word in the dictionary.

Find a longest monotonically increasing subsequence of $A[1..n]$.

$k$ painters to paint $n$ boards.

Given $A[1..n]$, find maximum sum of a subsequence of $A$, in which consecutive elements of $A$ cannot both be selected.

Similar to above, but subsequence is not allowed to have consecutive integers.

Given $A[1..n]$ of positive integers, negate some of them to minimize the sum of $A$, without making the sum negative. $\{2, 1, 4, 3, 2\} \rightarrow \{-2, -1, 4, -3, 2\}$.

**External problems**

Given a large file (orders of magitude 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.

Given a (really long) stream of characters, find the length of a shortest substring that has at least $k$ different characters, or the length of a longest substream that has at most $k$ different characters.

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.

**Games and puzzles**

Two elements of a binary search tree (BST) are swapped by mistake. Find the nodes and restore the tree.

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?

Egg-drop problem.