Problem solving session: Fall 2018

- Learning is not a spectator sport. Participate!
- Goal: discuss problem solving techniques. You can always look up answers to these problems on the web.
- Questions and discussions are welcome.
- For some problems, there may be better answers available by discovering additional structure of solutions.

Topics covered in interviews

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

- String algorithms: Knuth-Morris-Pratt (KMP) string matching, Boyer-Moore string matching.
- String data structures: Tries, Suffix trees (compressed tries).
- Bit manipulations.
- Finite State Machines, Regular expressions.
  - Programs with complex nested if statements can sometimes be simplified to a switch statement based on the state of a FSM, that is easy to code and easy to understand.
  - Ex: Rearrange 1→ 2→ 3→ ... to 1→ 4→ 7 ... 2→ 5→ 8...3→ 6→ 9...
- Number theory: primes, factorization, GCD, mod arithmetic.
- Enumeration: permutations, combinations with precedence constraints (e.g., how many topological orders does a given DAG have?), exploring all states of a game.

Traps to avoid

- 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^{19}), Chess (10^{45}).
- Bad ideas: greedy strategies, global variables
- Algorithms that are designed without consideration of their correctness, tend to be wrong.
- Harder the problem, easier it is to design a wrong algorithm!

Frequently asked questions

- When is the right time to apply for internships? Should I finish core courses first? Should I take easy-A classes first so that my GPA is high when I apply for jobs? Is it ok to graduate without doing an internship?
- Should I accept a company’s request to continue as an intern in the fall, after my summer internship? Will this affect my chances of getting a full-time offer if I say no?
- Is it a good idea to gain more experience doing internships, possibly delaying graduation?
- I have good ideas on how to solve problems, but I have difficulty finishing the coding task quickly. How do I enhance my coding skills?
- I am very stressed during interviews and do badly. What can I do to improve my performance?
- It doesn’t matter what I learn. All I need is a job when I graduate, right?
- What topics do job interviews cover? Is it enough if I study data structures and algorithms?
- What can I do besides taking courses to improve my job prospects?
• Given a word description of a problem, how do I recognize what algorithm to apply? Which graph algorithm should I use? Should I use dynamic programming?
• What are other attributes that help in being successful in interviews? Can they be developed to improve my chances? Points to ponder: cleanliness, body language, communication skills, networking skills.

**Subset sum and related problems**

Given an array of \( n \) integers, are there indexes \( i < j < k \) such that \( A[i] + A[j] + A[k] = 0 \)?

Find all unique triplets of the above kind.

What about arbitrary subsets of \( A \) with zero sum?

You are given a set of positive integers. The goal is to change the signs of some of the numbers and minimize the sum of all the numbers, without making the sum negative.

E.g. Set = \{ 2, 1, 4, 3, 2 \}. Minimum Sum is 0: \{ -2, -1, 4, -3, 2 \}.

**Knapsack and its variants**

**Subset sum**: Given a set \( S \) of positive numbers and a target \( t \), is there a subset of \( S \) whose sum is equal to \( t \)?

**Knapsack**: Given \( S \) and \( t \), find a subset with maximum sum, but \( \leq t \).

**Set partition**: Can \( S \) be partitioned into two subsets with equal sum?

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

**Weighted knapsack**: Given items \( a_1..a_n \), where item \( i \) has value \( v_i \) and weight \( w_i \), a knapsack of capacity \( K \), find items of maximum total value, whose total weight is less than or equal to \( K \).

**Bonus knapsack**

You are given items \( a_1..a_n \), where item \( i \) has value \( v_i \) and weight \( w_i \), a knapsack of capacity \( K \), and a set of bonus cards \( b_1..b_q \). Example:

\[ v = w = \{ 2, 2, 2, 2, 4, 4, 5 \}, \quad K = 10. \]

You can apply each bonus card to one item of your choice that you have selected. If a bonus \( b \) is used on item \( i \), then you obtain a value of \( b \times v_i \), instead of just \( v_i \).

In the example above, without bonuses, the optimal solution is 10, which can be obtained in many ways:

\{2, 2, 2, 2\}, \{2, 2, 2, 4\}, \{2, 4, 4\}.

If \( b = \{ 3 \} \), then the optimal solution is \{4, 5\} and the \( 3 \times \) bonus is applied on 5, giving a value of \( 4 \times 1 + 5 \times 3 = 19 \).

If \( b = \{ 2, 2 \} \), then the optimal solution is \{2, 4, 4\} with value 18.

**Card game**

Each card has a cost and a damage caused to your opponent. Given how much money you have and how much total damage you want to inflict on your opponent, and cost/damage information of cards \( C[1..n] \), find if it is possible to meet the goal within the available budget.

Example: money = 5, damage = 5, costs = [3,5,1] and damages = [2,2,3]. The goal can be achieved by buying \( C[1] \) and \( C[3] \) at a cost of \( 3 + 1 = 4 \), causing a damage of \( 2 + 3 = 5 \).

**kth largest element, median, selection problems**

**Internal versions**:
Given an array and \( k \), find its \( k \)th largest element.
Find the median of a given array.

**External versions**:
Given a stream of integers, find its \( k \) largest elements (say, \( k = 100 \)).
Find the median of an enormous array of integers, residing on a disk.
Reformulation of “Oversized Pancake Flipper”

Array of \( n \) switches, each “On” or “Off”. Any consecutive \( k \) switches can be toggled simultaneously. Find minimum number of steps to get all switches to On position.

Example: 

\[
\begin{array}{cccc}
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\( k = 3 \). Answer: 3 steps.

Approach for small values of \( n \), say \( n = 10 \)?

What about larger values of \( n \), say \( n = 1,000 \)?

Source: https://code.google.com/codejam/contest/3264486/dashboard

**Puzzles and games**

Sudoku, 8-puzzle, 15-puzzle, Magic squares

Minimum no. moves to win a snakes-and-ladders game with loaded dice.

**Word games**

Given a dictionary, find shortest number of steps to convert a word into another word, if in each step, one letter is substituted, and every word in the sequence must be a valid word: cat → can → wan → win.

Find a longest word that can be made to vanish by removing one letter at a time, if at each step you must have a valid word: sprint, print, pint, pit, it, i.

Break a given string into words from a dictionary.

Paragraph formatting problem: find places to break lines, and extra spaces to add to left and right justify all lines. A penalty function is specified that gives the penalty of a line depending on how many extra spaces are added. Goal: minimize total penalty of formatting the paragraph.

**Problems on trees**

Lowest Common Ancestor (LCA) of two nodes \( s, t \) on a rooted tree, where each node stores its list of children and its parent. Challenges: \( O(1) \) extra space, \( RT = O(d) \), where \( d = distance(s, t) \).

Diameter of a tree: find a maximum length path in the tree.

Weighted version: length of a path = sum of weights of its edges.

Challenge: Unrooted tree, given as a graph.

See also https://leetcode.com/problems/binary-tree-maximum-path-sum/description/

Given a binary tree that stores one element per node, is it also a binary search tree (BST)?

Conviviality rating (see Cormen et al, “Intro to algorithms”).

**Data structures**

Sort the elements of a linked list using only \( O(\log n) \) extra space in \( O(n \log n) \) time.

Challenge: Sort using only \( O(1) \) extra space, in \( O(n \log n) \) time.

FreqStack: a stack with operations push(\( x \)), pop(), and freqPop().

freqPop(): find the most frequently occurring element \( x \) on the stack, and remove and return the most recently pushed copy of \( x \). Break ties by selecting the element that was pushed most recently.

Source: https://leetcode.com/contest/weekly-contest-99/problems/maximum-frequency-stack/
Data structure to support add(x), contains(x), remove(x), getRandom().
getRandom() should select any one of the elements uniformly at random.

Hidden graph problems
Nuclear Rods are connected in groups. Input is given as pairs of connected rods such as follows:
{1,2},{5,10},{13,7},{7,6}. Find how many groups are there, and print the elements in each group: {1-2, 5-10, 7-6-13}.

Given a robot cleaner in a room modeled as a grid. Each cell in the grid can be empty or blocked. The robot cleaner can move forward, turn left, or turn right. When it tries to move into a blocked cell its bumper sensor detects the obstacle and it stays on the current cell. Design an algorithm to clean the entire room in O(n) moves, where the room has n cells.

Given n courses, and a set of m prerequisite pairs, find a sequence in which courses can be taken without violating prerequisite constraints.
Challenges: complete in min number of semesters, if taking up to 3 at a time. Complete just a given set of k courses and prerequisites needed.
You need to cross a stream by stepping on rocks that are jutting out of the water. You can jump at most a distance of d. Given the coordinates of all the rocks, find the least number of jumps needed to cross the stream.
Challenges:
In how many different ways can the stream be crossed if every jump must take you closer to the other side?
In how many ways can the stream be crossed without stepping on the same stone twice?

Given a set of money transactions, find the minimum number of transactions required to settle the debt.

Given a grid of numbers, find a longest chain of monotonically increasing numbers, where each element of the chain is one of 4 or 8 nearest neighbors of the next element in the chain.

Dynamic programming and enumeration problems
Find a largest non-decreasing subsequence of a given sequence.
Edit distance.
Given strings R, S, T can R and S be interleaved to make T?

How many subsequences of length k are there of \{1, 2, ..., n\}, if every element of a subsequence must be at least twice its previous element?

Count/enumerate all strings with balanced parentheses of n pairs of ()

In how many possible ways can we have a binary sequence of length n without 2 or more consecutive 0’s?
Egg drop problem.
Knapsack with limits: Find a subset of at most b elements with maximum total value, where the total weight of the chosen elements is at most K.
Two players play a game, starting with a row of n coins, with values v_1...v_n. Players take alternate turns, and in each turn, a player takes either the first or the last coin. Find the maximum total value that player 1 can make with perfect play.
**String problems**

Given a list of words (dictionary), and a grid of letters, find chains of letters forming words from the given list.

Giving a dictionary and a string, find the longest word in the dictionary that is a subsequence of the given string. Ex: “abpcplea”: “apple”

Given two strings $S$ and $T$, find a shortest substring of $S$ that contains all the characters of $T$. Ex: $S = ADBDCDBBANC$, $T = ABC$. Out: $BANC$.

Given a grid of letters and a word, find a chain of letters in the grid forming that word.

Given a string, find its longest substring without repeating characters.

**Bits and pieces**

Given an array $A[1..n]$ of binary numbers, find indexes $i\leq j$, such that toggling all bits $A[i..j]$ maximizes the number of 1’s in $A$.

Source: https://www.interviewbit.com/problems/flip/

Given two long integers $a$, $b$, find the minimum number of operations to convert $a$ into $b$. In each step, one of the following operations is applied.

a) $x = x\%2 === 0 ? x+1 : x-1$; (if even add 1, else subtract 1).

b) $x = x \times 2$; (multiply by 2).

A number is sparse if there are no two adjacent 1’s in its binary representation. For example 5 (101) is sparse but 6 (110) is not sparse. Given $x$, find the smallest sparse number $\geq x$.

**Subarray sum**: Given an array $A[1..n]$, find a subarray $A[i..j]$ with maximum sum.

Find a subarray $A[i..j]$ with minimum sum.

Find a subarray with zero sum.

Find a subarray with sum $k$.

Find subarray with sum as close to $k$ as possible.

**Misc**

Data structure and algorithm to merge $k$ sorted lists.

Given a regular expression and a sequence of strings, identify which strings match the regular expression.