Problem solving session: Spring 2018

- Learning is not a spectator sport. Interactive participation is crucial.
- Goal: discuss problem solving techniques. Understanding the techniques. You can always look up answers to these problems on the web. Take your own notes.
- Questions and discussions are welcome.
- For some problems, there may be better answers available by discovering additional structure of solutions.
- Topics to prepare beyond Algorithms (CS 6363) and Data Structures (CS 5343):
    - String algorithms: Knuth-Morris-Pratt (KMP) string matching, Boyer-Moore string matching, Tries, Suffix trees (compressed tries), Manacher's algorithm for longest palindromic substring
    - Finite State Machines
    - Number theory: primes, factorization, GCD, mod arithmetic
    - Enumeration: permutations, combinations with precedence constraints
- 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.

Knuth-Morris-Pratt (KMP) algorithm for string matching

- Find occurrences of a pattern $P[1..m]$ in a string $T[1..n]$
- A prefix function π is calculated for P in $O(m)$ time

$\pi(k) =$ Length of a longest prefix of $P[1..k]$ that is also a suffix of $P[1..k]$

a	b	a	a	b	a	b
0	0	1	1	2	3	2

- Using π, all occurrences of P in T can be found in $O(n)$ time
- When a mismatch occurs, slide the pattern by enough steps, so that, in the new position, the part of the pattern that overlaps the text is known to match
- If $P[1..k] = T[i..i+k-1]$, but $P[k+1] \neq T[i+k]$, then we set $k \leftarrow \pi(k)$. For this new value of k, we know that $P[1..k] = T[i..i+k-1]$ without making any comparisons, because by the definition of π, $P[1..\pi(k)] = P[k-\pi(k)+1..k]$, and we already knew that $P[1..\pi(k)] = T[i..i+\pi(k)-1]$

a	b	a	a	b	a	a	b	a	b
a	b	a	a	b	a	b			
			a	b	a	a	b	a	b

- **Boyer-Moore algorithm** for string matching: Text is scanned left to right, but for a given shift, pattern is matched with text from right to left. Two heuristics are used to decide new shift when a mismatch occurs: bad character rule, good suffix rule. Boyer-Moore algorithm performs better than $O(n)$ for many inputs, because the bad character rule may propose big shifts, and many parts of the text are never compared with the pattern.

- **Trie**: a simple data structure for searching for keywords (strings from a dictionary). Used for searching keywords, dictionaries, address books.
- **Suffix tree**: a compressed trie built with all suffixes of a string.
- **Manacher's algorithm**: find a longest palindromic substring of S in $O(|S|)$ time.
- **Finite state machines**: a machine with a fixed number of states that can be used to model problems. Programs with complex nested *if* statements can sometimes be simplified to a *switch* statement that is easy to code and easy to understand. Example: Rearrange 1→ 2→ 3→ 4→ … to 1→ 4→ 7 ...2→ 5→ 8...3→ 6→ 9…
- **Enumeration**: Permutations, combinations with precedence constraints (e.g., how many topological orders does a given DAG have?), exploring all states of a game

Dynamic Programming strategies

- Find a solution using divide and conquer. Identify all subproblems that arise when problem is solved. Design storage to store all these solutions. Generate problems in order of increasing sizes (i.e., the induction order), compute their solutions without using 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 about the case when 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 that are designed without consideration of their correctness, tend to be wrong. Harder the problem, easier it is to design a wrong algorithm!

DP: Box Stacking.

You are given a set of n types of rectangular 3-D boxes, where the i^{th} box has height $h(i)$, width $w(i)$, and depth $d(i)$. You want to create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the dimensions of the 2-D base of the lower box is strictly larger than the 2-D base of the upper box. Of course, you can rotate a box so that any side functions as its base. It is also allowed to use multiple instances of the same type of box.

DP: Max subsequence sum

Given an array of integers find maximum sum of a subsequence, such that if you include a number x, then you cannot include any elements equal to $x-1$ or $x+1$.

Eg, if numbers are 2 1, 2, 1, 2, 3 then the maximum sum is including the 2's, and excluding 3's and 1's.

Assume that $n \le 10,000,\ 1 \le arr[i] \le 10,000,\ i = 1..n$.

Tree: Inorder successor in a BST

Find the inorder successor of a TreeNode x in a binary search tree:

class TreeNode { int val; TreeNode left, right; TreeNode(int x) { val = x; } }

Game: Word maze

Given a collection of words, find them in a given matrix of letters, where each word is formed by adjacent letters on a line (horizontal, vertical, or diagonal).

Game: Change signs to minimize sum

You are given a set of positive integers. The game 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 }.

String/List

Given two linked lists of strings, each in lexicographic order, output a list of strings that are in both lists (intersection of two sorted lists).

Tortoise and hare

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 it be solved with only one iterator? Can it be solved without storing address of intermediate nodes?

Data structure

Design data structures to support the following operations:

add(x): add x to the data structure (duplicates allowed),

median( ): return the median of the elements added so far.

What if remove(x) operation is added?

External: Unique elements

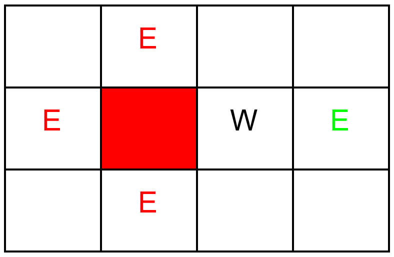
Given a large file (orders of magnitude bigger than available memory), where each line has one or more integers, find the lines of the file that are distinct. Two lines are considered to be the same if they contain the same integers (possibly in different order). For example, "1 2 3 3" is the same as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

Sorting objects by color

Given n objects colored 0, 1, 2, sort them by color. E.g., Input = { 2 1 2 2 0 1 0 }, Output = { 0 0 1 1 2 2 2 }. Solve the problem in $O(n)$ time in a single pass, using at most $O(1)$ extra space.

Puzzle: Plant-a-bomb

Given a 2D grid, each cell is either W (wall), E (enemy), or blank, find a blank cell to place a bomb that kills the maximum number of enemies. A bomb kills all enemies in the same row and column as the bomb, provided there is no wall in the way. E.g.: in following example, placing a bomb at (1,1) kills 3 enemies.



Puzzle: 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.

```
  *       * * *     | * *
* X *     * X *     * X |
  *       * * *     * * *
```

String/DP: Given a dictionary of words, and a set of strings, write a program that decides if each string can be broken into substrings that are words in the dictionary (with as few words as possible).

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 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?

Selection problem:

Internal version: Given an array and k, find its k^{th} largest element.

External version: Given a stream of integers, find its k largest elements (say, k=100). Assume that k is smaller than available memory.

Array: Find max

Given a bitonic array, find the point of inflection. An array is bitonic if it is made of a monotonically increasing subarray followed by a decreasing subarray.

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?

Build a web server for looking up meanings of a dictionary of words (say, 100,000 words, with average length of around 5 characters) and meanings of these words.

Given a phone number, break it into words from a dictionary (e.g., 1-800-FLOWERS for 1-800-356-9377). Mapping: 2=ABC, 3=DEF, 4=GHI, 5=JKL, 6=MNO, 7=PQRS, 8=TUV, 9=WXYZ.

Dynamic dictionary, with operations add, remove, and find.

Look up a contact stored in phone's address book by name (or part of name?)

String: Longest substring with 3 distinct characters

Given a string, find a longest substring that is composed of at most 3 (in general, k) distinct characters. Same problem on a long stream of characters. What about a shortest substring that is composed of at least k distinct characters?

Shortest subarray that contains all elements of a given array.

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

Array: Longest streak

Given an unsorted array of integers, find length of a longest streak of consecutive integers. Example: {1,7,3,2,9,4,8}. Output: 4, corresponding to the streak {1,2,3,4}.

Priority queue: Perfect power

A number if called a *perfect power* if it is equal to a^b, for some integers a and b. Write a program to enumerate all perfect powers between 2 and 10^8.

Array: Maximum sum

Given $A[1..n]$, find maximum sum of a subarray of A. Subsequence? Monotonic subsequence? Subsequence that does not contain any two adjacent elements?

Graph: Number of paths

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

String: Palindromes

Longest palindromic substring.
Longest palindromic subsequence.
Longest subsequence that forms a balanced set of parentheses.
Break a string into smallest number of palindromes.
Count number of palindromic subsequences.

Tree: Reconstruct from traversals

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

Graph: 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?

Array: 2-player coin game

Given a row of coins 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. 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?

Scheduling: Group interval scheduling

Given groups of activities, $G_1 \ldots G_k$, where each group has 2 activities, check if there is a collection of k compatible activities, one from each group.

Tree: Level order traversal

Given a binary tree, return a list of lists of its nodes (one list for each level).

DP: Offline 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.

Archaeology: Alien language

You come across a new alien language and a dictionary of words from that language. Find the alphabetical order of the letters in this language.

Ex 1: { wrt, wrf, er, ett, rftt } → Alphabetical order: { w, e, r, t, f }, Ex 2: { zxx, xxz, zzz } → Invalid.

String: subsequence count

How many times does T occur as a subsequence of S?

Ex: S = accbcacbc, T = abc. Answer: 4.

acc**bc**abc, **a**cc**b**cab**c**, **a**ccbca**bc**, accbc**abc**

String/Data structure

Distinct substrings: Find the number of distinct substrings of a given string.

Repeating prefix: Find longest prefix p of a string which has pp as a prefix.

Search: Given a dictionary of words, searching by substring.

Superstring: Shortest substring of S that contains T as a subsequence. Assume T is a subsequence of S.

Game: Snakes and ladders: Find the minimum number of moves to win a game of snakes-and-ladders.

String: Lexicographically smallest distinct subsequence

Given a string (array of characters) of letters (a-z), remove duplicates of each character such that the resulting string is the lexicographically smallest string.

Example: cbbadcdcac Answer: bacd.

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.

Examples: aacecaaa → **a**aacecaaa, abcd → **dcb**abcd

Game: Word alchemy

Given a dictionary of words, preprocess it to answer edit distance queries, where in each step one letter is replaced.

Convert(hit, cog): hit → hot → hog → cog.

Convert(toad, pond): toad → told → bold → bond → pond.

Implementation of LRU cache

Cache stores n objects, based on key. Operations:

get(k): return object with key=k, if it exists in cache, null otherwise.

put(k, x): place x with key=k in cache, replacing the least recently used object.

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={1,4,4,5,5,5,5,8,8,9,9}, 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.

DP: LMIS

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

Graph: Partitioning a tree

Given an undirected graph that is a tree, with weights on its vertices, find an edge to delete that results in 2 trees whose total weights are equal (or as close to each other as possible).

DP: 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 to find the maximum floor from which an egg survives a fall.

Given f floors, n eggs, find the fewest number of trials needed to find the answer.

Given n eggs, t trials, find the maximum number of floors that can be explored.

Array: Duplicates

Given an array A[1..n] of numbers, find if elements are distinct.

What if you can use only O(1) extra space outside the array?

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

Recursion

Convert a doubly-linked, circular, sorted list into a balanced binary search tree (prev=left, next=right) in $O(n)$, without allocating extra space.

Reverse problem: BST $\rightarrow$ Sorted list.

Recursion: Build a balanced binary search tree from a sorted array.

Game: 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.

Search a sorted matrix

Given an $m \times n$ matrix of integers, in which each row and each column is sorted, write a function to find an element x.

DP: Balanced subarrays

Given an array of letters and numbers, find a longest subarray with an equal number of letters and numbers.

Given a string containing just the characters \`(' and \`)', find the length of the longest valid (well-formed) substring. For \`\`(()'', the answer is \`\`()'', with length 2. For \`\`)()())'', the answer is 4.

Given a string, find a shortest extension that makes it well formed.

External sorting

Given a large number of short strings, and a distributed system (cloud), how do you sort the strings?

DP: Factorize to single digit

Given N, find the shortest number of steps to make it into a single digit number, if in each step, N is replaced by max$(x, N/x)$ for a divisor x of N, or by $N - 1$. Ex: 50850 $\rightarrow$ 226 $\rightarrow$ 225 $\rightarrow$ 15 $\rightarrow$ 5.

Game: Zuma

Balls of 5 colors. Initial configuration: k balls on table, n_i balls of color i on hand, $i = 1..5$. In each step, player inserts a new ball at any position on the table, and 3 or more adjacent balls of the same color are removed, repeatedly. Can the player play a sequence of moves to clear the table?