COP 4516 Spring 2024 Final Individual Contest Solution Sketches

Problem A: Slow Typing

This is the easiest problem in the set. First to get Arup's current typing rate, multiply his current rate by the fraction $f^2/100$. This will give his current typing rate per minute. Then take the total number of lines of code and divide it by this adjusted typing rate. Make sure to output to exactly 2 decimal places in whichever language you are using.

Problem B: The Grand Social Gathering

We require a Binary Search Tree (BST) to store names along with their corresponding frequencies, sorted by the name. To achieve this, we can employ a TreeMap of a string representing the name to the frequency. When we insert a name, we check to see if it exists in the map. If not, create a new entry for it, mapping it to 1. When we delete a name, we check to see if that name appears exactly once currently. If so, we simply remove this entry from the map. Otherwise, we map this entry to its previously mapped entry minus 1. (Note: Even if we mapped a name to 0, this would still work.) For queries, just use the built in function which provides the subsequent key in the map.

Problem C: Sum of Divisors

In class the formula for sum of divisors was given. Specifically two forms of the formula were given. Let $n = p^a q^b$, where p and q are distinct primes, and a and b are non-negative integers. (The formula extends to an arbitrary number of distinct primes in the prime factorization but without equation editor, it's easier to just limit this write up to two to get the point across.)

$$\text{sumD}(p^a q^b) = (1 + p + p^2 + p^3 + \ldots + p^a)(1 + q + q^2 + q^3 + \ldots + q^b)$$

Alternatively, one could use the geometric sum formula and simplify each of those sums to get:

$$\text{sumD}(p^a q^b) = \frac{(p^{a+1} - 1)(q^{b+1} - 1)}{(p - 1)(q - 1)}$$

Due to the smaller bounds on this problem (there are only 10 terms AND each exponent is no more than 100,000), it's feasible to implement the first formula (ten loops of length 100,000) instead of the second. Since the answer has to be reported under mod, to properly implement the second formula, mod inverse is needed. Since there is no division in the first formula, no mod inverse is needed. Also, as a side note, since 10^9+7 is prime, we can also prove that a number's modular inverse mod this value is simply that number raised to the $10^9 + 5$ power. This proof follows from the result of Fermat's Theorem (which was not shared in the class, but is typically covered in Number Theory and Cryptography courses.)

Finally, when computing the sum, a double for loop can't be used. One can not compute each term of the form p^i via a form loop upto i. Instead, one must notice that after we calculate p^i, we can store that in a temporary variable and then multiply that temporary variable by p once to obtain p^{i+1}.

Problem D: The Island of Valcu

Use BFS to find the shortest distance from each volcano to all cells on the island. Initialize a queue with the coordinates of all volcano cells, setting their distances to 0. This is much faster than running k BFSs, when there are k volcanoes.

In one of the judge solutions, once the distances from all volcanoes are calculated, another BFS traversal from the starting point towards the rescue zone is done. During this traversal, the algorithm updates the minimum distance from each cell to a volcano, ensuring that it maximizes the distance from volcanoes while traversing towards safety. This modification of a BFS is different than a regular BFS with a used array. You do allow visiting a square more than once (no used array). If an alternate visit to the square improves the minimum distance from a volcano to that point, we update that information. By the time the BFS queue is empty, when reaching the end square, it will store the maximum possible minimum over all possible paths.

In the other judge solution, after the initial multi-source BFS, each grid square is sorted in reverse order by distance from a volcano. Treat each grid square as a disjoint set, with each square starting in its own set. Traverse through the sorted list in order. When getting to a cell, look at its adjacent neighbors. If those neighbors are an equal or greater distance from a volcano as the cell itself, union these two cells in the disjoint set. At the exact point in time where both the start and end square are in the same disjoint set, then last union that created this, look at the distance of that square and that is the answer. (The idea here is that we only traverse on squares one by one from highest to lowest distance. As soon as the start and end are in the same disjoint set, that means there's some path that only uses the squares that have been unioned so far. The last union has the desired distance from a volcano.)

Problem E: Company Tug of War

In many ways, this is a traditional tree problem. The solution involves recursively calling the solve function on each subtree. Obviously, the maximum of these subtree answers is a candidate for the final answer. In addition, we must consider the root node hosting the tug of war. Notice that there are at most 10 children for any node. This means that we can simply run a brute force solution which evaluates all at most 2^{10} subsets of subtrees and simply pick the partition that creates the smallest absolute value difference in sums of weights. If this difference is larger than the maximum of the subtree answers, we return it. Otherwise, we return the maximum of the subtree answers.

To make sure inefficient adding isn't occurring, store the sum of all values in a subtree in each node, so O(1) lookups will find the required subtree sum of weights. If we do this, the run time should be good enough because there are no more than 10,000 nodes and we never do more than 1024 work per node, so the number of simply operations is roughly no more than about 10^7, and in reality it's much less than this because many nodes in the tree must be leaf nodes (it can be proven that this is at least roughly half the nodes because every node with children has at least 2 children).