Editorial – Ode de Code

- Ankit Kumar Vats

Problem: Ishant Can Not Code

Difficulty: Cake Walk

Prerequisite: Nothing

Just assume that the first number is the minimum one and compare the rest four numbers with it one by one. If any one of them is smaller than the current minimum, then make that minimum. Please see the attached solution.

Problem: Stan Swaps Glasses

Difficulty: Cake Walk

Prerequisite: Nothing

Store all the strings in an array of N+1 elements at the indices 1 to N. Then take another array A of N+1 element and initially make all elements as A[i]=i. They denote the indices of the strings present at that location currently. Now instead of swapping the whole string in each turn, just swap the indices. Please look at the solution for implementation.

Problem: Special Blessings to Players

Difficulty: Easy

Prerequisite: Hashing

If there are "N" players then take an array M of size N+1 with 0 indexing. If the current operation is "A B K", you just have to perform these two operations:

M[A]+=K
M[B+1]-=K

This is sufficient to find the solution in O(N)

After this initially take a variable var=0 and print the solution as follows:

```
for(i=0;i<N;i++)
{
        var+=M[i];
        printf("%d\n",var);
}
```

Problem: Dividing Money

Difficulty: Easy

Prerequisite: Dynamic Programming

Here you have to divide the cheques into two sets such that the difference of sums of the two sets is minimum. We take an array A of 10^5 elements (maximum possible sum of all the elements).

Make A[0]=1;

After this we have to make all those values as 1 which are possible to be obtained as sum of any subset.

Wo do this for all the cheques. Lex "X" be the amount of money in a particular cheque.

```
for( j = sum; j-X>=0 ; j-- )
        if(A[j-X]==1)
                A[j]=1;
```

After this we have to find the index for which A[index]=1 and is closest to sum/2.

Please look at the solution for details

Problem: Ross Generates Data

Difficulty: Easy **-** Medium

Prerequisite: Hashing

Suppose we have the array A. Now, the most challenging part of this problem is the large n. To solve this problem, you should notice 2 things. The maximum result of A[i] (i >= k) will not be larger than k+1, so even though A[i] (i < k) may be as large as 10^8, the solution is never larger than k + 1. Also, the result of A will be repeated every k+1 numbers. This means that we just have to calculate the k+1 next numbers, even if n is very large. Now we've reduced the problem to the following: find A[k], A[k+1], ..., A[2k+1].

The brute force version of doing this is still too slow (O(k^2)), so we have to use either a hash table or a BST to maintain the 'available' values. Please see my solution using hash tables.

Problem: Cartman and Butters Game

Difficulty: Medium

Prerequisite: Game Theory

We can solve this using basic Grundy and NIM theory. We can pre-compute the grundy values of all possible states the game can be in and then for each testcase, xor the corresponding grundy values and if the result is non-zero, Cartman (first player) wins, else Butters wins. First thing to note is, $Z \leq N$, as allowing a player to remove more stones than actually present is useless, so Z = minimum(N,Z). Let G(N,Z) be the grundy value of the state where N stones are present in the pile and at most K stones can be removed. If we remove $1 \leq p \leq K$ stones in this step, the resultant state is G(N-p,p). We have the following recurrence,

G(N,Z) = mex{ G(N-1,1), G(N-2,2), G(N-3,3), . . . , G(N-Z,Z) }

But this doesn't look good, as it may take O(N) time to find each of the all N^2 grundy values, leading to $O(N^3)$ time. If you visualize this on the grid, G(N,Z) and G(N+1,Z) are closely related.

G(N,Z+1) = mex{ G(N-1,1), G(N-2,2), G(N-3,3), . . . , G(N-Z,Z), G(N-Z-1,Z+1) }, depends on just one more additional value than G(N,Z).

We can find the grundy values of each row of G by maintaing the same boolean visited array and values may only increase as we move left to right. This can be done in O(N) time per row, so the overall complexity is O(N*N). Draw a grid on paper and see how this works.

This problem was not originally designed by me. It is taken from here:
http://www.codechef.com/COOK18/problems/TAKEAWAY

Problem: Help Walter White

Difficulty: Hard

This problem was also not originally designed by me. It is taken from here:

http://codeforces.com/contest/377/problem/E

You can find its editorial here: http://codeforces.com/blog/entry/10157