Editorial for Intra RUET Junior Contest (Round 1)

Happened on 7th August 2020 5:00 PM (+0600 GMT)

Descendant of the Sun

Author: Emama Nahid

This problem is really easy to solve. We need to take 2 strings as input and count the vowels from each string. Be careful about counting both upper cases and lowercase vowels.

Let C_1 be the count of vowels of the first string S_1 and C_2 the vowel count of the second string S_2.

If (C_1 > C_2) print First String.
If (C_2 > C_1) print, Second String.
If (C_1 = C_2) print, Equal.

GIFT

Author: Rezwan Ul Haque

After giving exactly **one pair** of integers to Ikri there remains a total of N -1 pairs of integers.

We are said to find a way by taking exactly one integer from each of N-1 pairs of integers that make the sum of all the taken elements equal to M.

We can solve this easily by using dynamic programming.

Just maintain a dp array, $DP_{i,j,k}$ [Assign -1 to every element of the array except $DP_{0,0,0}$ = 0 (why ? Find yourself)]

where i is the position of the pair we are calculating now, j is the number of elements taken so far, and k is the sum of the chosen elements.
But this won't be accepted because this needs huge memory.
Look, we have to delete only 1 pair. So, we can change the definition of j in our $DP_{i,j,k}$ array, where j $(0 \leq j \leq 1)$ is the number of elements we didn't take. i and k remain like before.
Now, just check everything as we do in knapsack.

$DP_{i,j,k}$ is equal 0 if we don't take any integer, 1, if we take the first integer or 2 if we take the second integer from the i-th pair, and the summation we have made so far, is k.
$DP_{i,j,k}$ remains -1 if there is no way to make summation equal to k by taking total (i-j) integer(s) from the first i pair(s).
If $DP_{N,1,M}$ = -1 then the answer is "NO", otherwise the answer is "YES". Now, just print the way you take the integer.

Complexity: O(N*M*T)
Here, is the author's solution: https://ideone.com/zj7UhX
Recursive DP solution: https://ideone.com/mENQuN

Yet Another Travelling Problem

Author: Nahid Hasan

In this problem, we need to find the $K-1$ number of consecutive edges of any type of edge from any vertex.

For this, we will build graphs for each type of edge. As no cycle won't contain the same type of edges, this graph will always be a forest. Forest is one kind of graph whose connected components are trees. So this forest will never contain any cycle (none of its components too). Now, from this graph, we just need to find the maximum depth among all the connected components (trees) that is nothing but finding the longest path AKA diameter. You can do it with two DFS/ BFS. Then compare this with the given K and output answer accordingly.

Asymptotic: $\mathcal{O}(N+M)$

Judge Solution: https://pastebin.ubuntu.com/p/RKTPvQk2dG/

Bad Story

Author: Abdus Salam

The problem is straightforward. You need to find the sum of elements that contains at most X number of divisors.

As the input is huge (10^6), sqrt factorization will give TLE. So we need to pre-calculate the number of divisors of all the numbers till 10^6 in O(nlogn). You can do it with the Sieve of Eratosthenes.

Let's define, $\text{divisor}[x]$ = number of divisors of x

Then for $i-$th $(1 \leq i \leq N)$ element let's build an array sum initialized with the value 0 (zero).
Then for each element a_i, build the array in the following way,

$$\text{sum}[\text{divisor}[a_i]] += a_i$$

Next, we need to build another array that will be the prefix sum of the sum array.

$$\text{prefixsum}[i] = \text{prefixsum}[i-1] + sum[i]$$

Then for each query, you will be given X and the answer will be $\text{prefixsum}[X]$.

Asymptotic: $\mathcal{O}(N \log N)$

Judge Solution: https://pastebin.com/LrcLGc7S

Let's Make String Interesting

Author: Golam Rabbi Nazum

We can naively try to set unique characters optimally.

We will try to set X unique characters in the modified string. Then X should divide the length(N) of the string. So that each of the X characters occurs exactly N/X (equal) times.
Among all of the possible valid values of X, we will set that value where we will need a minimum number of operations.
So the valid values of X are the divisors of the length of the string which is less than or equal to 26.
Because there are only 26 unique characters possible.

Now the question is: What is the optimal way to set a fixed valid number of unique characters?
The answer is: Find it by yourself.

Authors solution (C++): https://ideone.com/lIlpz5
Complexity : $\mathcal{O}(\max(26 * 26, N))$

The Queen

Author: Joytun Sultana

In every number system, we can count sequentially. But we represent them using different digits. For example, we represent 4 in binary as 100 and 5 as 101.

In this problem, we have considered 0 as our 1st number or at position 1. So we can say that "**position= number+1**". As we started our number from 0 but the position from 1. This is true for the **base-10** number system. So if we convert our given number N into **base-10** then the condition "**position= number+1**" will be satisfied.

But the problem is we don't know the actual base of the given number N, which is necessary for converting N into **base-10**. Let's define this unknown base as y. Convert the given (N_y) into (N_{10}).

We know that for our problem y (base) can be from 2 to 10.
So we have to convert that given number N into base-10, for every possible y from 2 to 10 and whenever the condition "**position = number$_{\text{(base 10)}}$+1**" satisfies, we can say we have found a valid base for that given number.
If "**position = number $_{\text{(base 10)}}$ +1**" not true for any base then there is no solution and print (-1).

"Remember that any digit of a number is less than its base"

Judge Solution: https://pastebin.com/MEGG2fQD
Judge Solution (2): https://pastebin.com/w8NM09As

Treat or Trit?

Author: Tahsin Protik

Let's determine the sum of every unique pair in a particular range.
Firstly, we will calculate the sum trit-wisely. We will count the number of 0, 1 & 2 in i-th bit.
Let's look at this,

1. $(0+1)\%3=1$

2. $(0+2)\%3=1$

3. $(1+2)\%3=0$

4. $(1+1)\%3=2$

5. $(2+1)\%3=1$

If we have a close look, we can say that, if we have the number of 0, 1, 2 in i-th bit, we can calculate the number of pairs that have the positional value of 0, 1, 2. Then we calculate the original value by multiplying them with 3^i. Then we will sum up this for every Trits.
We can use a segment tree for calculating the number of 0, 1, 2 in any rage for every Trit.

Complexity: $O(3 \times N \times log2(N) \times log3(10^9)$

Judge Solution: https://paste.ubuntu.com/p/RqqCgDTYNM/