ProCo 2019 Round 2 Solutions

Ali(p)en Attack (100)

The answer is encoded in PigPen, which is a pretty popular cipher:
ctfisinevitable

Substitute the Wire (200)

This is a Vignere Cipher, but it was constructed such that any online decoder could use statistical analysis to solve it (or if you were bold enough to write the Kasiski method yourself).

The flag is located at the end of the file: **palladiumtobadassium**

Custom Language (300)

We give you an alphabet: poiuytrewqmnbvcxzlkjhgfdsa0987654321

Use this alphabet and bruteforce a ROT (or shift) to get: **m1cha3lr00k3risy0ndu**.

セマフォル (300)

This is Japanese and sounds like semaphore…

Turns out the cipher is made up of semaphore flags, which if you decipher becomes: **stevelovespeggy**

Asteroid Attack (200)

If you look around, you have to sharpen the image in the middle of space to see the flag: **str4nd3d_1n_sp4c3**

Location: South of Boot-2 (500)

The title leads to the clue: LSB-2 where we take the least significant 2-bits to encode the information. Deciphering it, we get morse code which then translates to: **g0dofhamm3r5**

Thanos (200)

We use only the letters in ‘thanos’ to encode the 9-length password: **sntohsnat**

The solution is to bruteforce the md5 hash until you get a match.

Searching for the Alpha, Bravo and Charlie (300)

We put NATO alphabet words in the Lord of the Rings Text. We tried (hopefully) to mask the difference between the text so you couldn't just diff the files online and the current one to find readable text. If you just filter on the NATO Alphabet words you should spell out: **WHEREDIDTHEYGO**

Piecing the Titles Together (400)

We created bigrams from the titles. Try to anagram them together to get the corresponding titles and then the first letter of each title becomes:
ATTACKTHANOSATDAWN

Love (100)

The provided function takes a number and returns the number of divisors of the number.

Inspecting the list, we see that all of the numbers have 32 factors.

We must find the 20th smallest positive integer with 32 factors.

Answer = 3000

Time (400)

T function: for a given number num, returns number of ordered pairs (a, b) of positive integers such that $a^2 + b^2 = num$.

I function: for a given number num, returns the number of zeros in its base 10 representation.

M function: for a given number num, returns the number of digits.

E function: for a given number num, returns the sum of digits.

Time (400)

We want odd 8-digit numbers with 4 zeros and whose digits sum to 16.

Also, there must be 12 ordered pairs of positive integers (a, b) such that $a^2 + b^2$ equals our number.

Since our number must have 8 digits, a, b must be less than 10,000. If we keep an array A, we can loop through all possible a, b instead and increment $A[a^2 + b^2]$. Then we can loop through A to check whether its value is 12 and the other properties of the number are satisfied.

Time (400)

12 possible numbers:

14000605, 27000025, 36000025, 40010605, 40050025, 40100065,

54020005, 60003025, 60104005, 60300025, 70000225, 81000025

Loop (200)

Hash function for string $s_0s_1...s_{n-1}$: $(2^0*H[s_0] + 2^1*H[s_1] + ... + 2^{n-1}*H[s_{n-1}]) \bmod M$

Hash of string “theavengers”: $X = (2^0*H[t] + 2^1*H[h] + 2^2*H[e] + 2^3*H[a] + 2^4*H[v] + 2^5*H[e] + 2^6*H[n] + 2^7*H[g] + 2^8*H[e] + 2^9*H[r] + 2^{10}*H[s]) \bmod M$

Hash of string “theavengerstheavengers...theavengers”:

$X*(2^0 + 2^{11} + ... + 2^{11(n-1)}) = X*(2^{11n} - 1)/(2^{11} - 1) \bmod M$

Loop (200)

Hash of string “thanos”: $Y = (2^0*H[t] + 2^1*H[h] + 2^2*H[a] + 2^3*H[n] + 2^4*H[o] + 2^5*H[s]) \bmod M$

Hash of string “thanosthanos...thanos”:

$Y*(2^0 + 2^6 + \ldots + 2^{6(n-1)}) = Y*(2^{6n} - 1)/(2^6 - 1) \bmod M$

Loop (200)

We want $X*(2^{11n} - 1)/(2^{11} - 1) = Y*(2^{6n} - 1)/(2^{6} - 1) \bmod M$ where $M = 14000605$

Since the map *H* is unknown, the above needs to be true for all X and Y. The only way to guarantee this is if both sides are zero modulo M.

Since $\gcd(M, 2^{11} - 1) = \gcd(M, 2^{6} - 1) = 1$, we need $2^{11n} = 2^{6n} = 1 \bmod M$. This is equivalent to $2^{n} = 1 \bmod M$.

The smallest positive integer n that works is 164696.

Endgame (200)

The function $f(x) = 512x + 2019 \bmod 14{,}000{,}605$ is periodic.

If $f(x) = 1$ for some x, there exists some positive integer n such that $f^n(x) = 1$.

We can find the cycle that 1 belongs to (length is at most 14,000,605), reverse the cycle, and index into the cycle to find the answer.

The Stack Laptop Part A (200)

Let x denote the value pushed in the stack.

After the first loop, the stack will be [x, x-1, …, 2, 1, 0].

The second loop sums them up, so result = x(x+1)/2.

x(x+1)/2=8590000128; solve for x.

x = 131072.

The Stack Laptop Part B (300)

Let x and y denote values for the first and second PUSH instructions.

In the first loop, stack will be filled with powers of 2, if stack[top]=2x-2, FUNCT is called. FUNCT simply inserts stack[top]+1 below stack[top].

The first loop ends when stack[top]=2^{y-2}.

The second loop adds everything up. So result=1+2+...+2^{y-2}+(2x-1)=2^{y-1}+2x-2 if 2x-2 is a power of two smaller than 2^{y-2}.

x, y = 32769, 34 so report 3276934.

The Infinity Laptop Part A (200)

For a given initial state, using these instructions, it is possible to produce every permutation of stack[top], stack[top-1] and stack[1], e.g. for n=3 all 3!+3*2!+3*1!+0!=16 states are possible.

Can write a recursive function to count the number of states, POP one element and call itself.

Can keep track of the set of all states to prevent double-counts more easily.

The answer is 821.

The Infinity Laptop Part B (300)

For a given initial state, using these instructions, it is possible to produce every permutation of stack[top], stack[top-1], stack[top-2] and stack[1], e.g. for n=4 all 4!+4*3!+6*2!+4*1!+0!=65 states are possible.

Can write a recursive function to count the number of states, POP one element and call itself.

Can keep track of the set of all states to prevent double-counts more easily.

The answer is 5861.