Rusty String

Problem Summary: Given a string consisting of V, K, ? where ? represents an unknown value find all possible periods among all possible strings

- First glance this looks like a string problem. And if all characters were known we can use the Z algorithm to find the answer in $O(n)$

- obviously we can't generate all possible strings since there is $O(2^n)$

- Lets assume a period of K is possible then $A = \underbrace{V\_\_\_\_}_{K}\underbrace{V\_\_\_\_}_{K}\underbrace{V\_\_\_\_}_{K}\dots\underbrace{V\_\_\_\_}_{\le K}$ then all positions $A[0] = A[K] = \dots A[N \cdot K]$ but we could have $A[0] = ?$ and $A[K] = V$ and still be valid, so we need a better method to check

- A period of K generates K groups element A_i is in group $i \% K$
  eg. If $N = 10$ and $K = 4$ $g_1 = \{0, 4, 8\}$ $g_2 = \{1, 5, 9\}$ $g_3 = \{2, 6, 10\}$ $g_4 = \{3, 7\}$

- Then period K is valid iff for all elements in each group both V & K are not in the group. From this property we can check all groups in $O(n)$ time for total $O(n^2)$. Good but still not good enough

- observation: Lets say we have a string containing both V and K so $A = \dots\dots \underbrace{V \dots\dots K}_{d} \dots\dots$ and their distance is K then a period of d isn't possible

- Stronger observation: If we have V & K at a distance of d then all factors of d are not valid periods. This is because $\underbrace{V \overset{m}{|\,|\,|} K}_{d}$ since d is divisible by m both V & K will be in the same group if m was the period

- so if we find all distances between Vs & Ks we are done. Naively doing this takes $O(n^2)$ time. Still not better. Represent all positions of Vs as i and all Ks as j then $d = |i - j|$ to compute for all i & j represent as two polynomials $(x^{i_1} + x^{i_2} + x^{i_3} \dots x^{i_n})$ and $(x^{-j_1} + x^{-j_2} + \dots + x^{-j_L})$ if we multiply these two polynomials then the ds are incoded as coefficients

- To multiply these two we can use FFT to do this in $O(n \log n)$

- Now we just iterate over all d values and their factors. This is straight forward in $O(\sqrt{n} \cdot n)$ with $N = 5 \cdot 10^5$ $\sqrt{N} \cdot N \sim 3 \cdot 10^8$ which ACs with a good implementation

- Follow up: $O(n \cdot \log n)$ is possible try to optimize the last step