Coding Practice With Kick Start 2020 - Coding Practice with Kick Start

Analysis: Big Buttons

Test Set 1

Since we have 2 choices at every step, there are $2^{\mathbf{N}}$ possible strings of length **N**. We can generate all of these strings in $O(2^{\mathbf{N}})$ time using a naive recursion. For each string, we need to check if any of the **P** strings is its prefix. A simple implementation of the above would run in $O(2^{\mathbf{N}} \times \mathbf{P} \times \mathbf{N})$ time for each case. Since $\mathbf{N} \le 10$ in this dataset, this is fast enough.

Test Set 2

In this dataset, since **N** can be as large as 50, it is not feasible to generate all $2^{\mathbf{N}}$ strings. The main idea in solving Test Set 2 is to find the number of invalid strings of length **N** and subtract it from the total number of strings ($2^{\mathbf{N}}$).

If a forbidden prefix has length L, we have $2^{(\mathbf{N} - L)}$ invalid strings of length **N** with that prefix. Also, if we have two strings A and B, and A is a prefix of B, then all strings that have B as their prefix will also have A as their prefix. So we can remove all forbidden prefixes that themselves begin with a different forbidden prefix to avoid overcounting the sum of invalid strings.

We can now easily count the sum of invalid strings by finding the number of invalid strings for each remaining forbidden prefix. We can remove redundant forbidden prefixes in $O(\mathbf{P} \times \mathbf{P} \times \mathbf{N})$ time, and find the counts of strings with the remaining forbidden prefixes in $O(\mathbf{P} \times \mathbf{N})$ time, so the overall running time is $O(\mathbf{P}^2\mathbf{N})$. This is fast enough to solve Test Set 2, but can you see how to improve the first step to $O(\mathbf{P} \times \mathbf{N})$ time by using a trie?