Kick Start 2018 - Round E

Analysis: Milk Tea

For the Small dataset, there are only $2^{\mathbf{P}}$ different combinations of options available. Since **P** is at most 10, there are only 1024 possible combinations. Take the combination that gives the fewest mistakes that isn't forbidden.

For the Large dataset, first notice that there are at most 100 forbidden combinations. One approach is to generate the 101 combinations that cause the fewest complaints and take the best one that is not forbidden (there is at least one combination that is not forbidden in the top 101).

To generate these combinations, begin by noticing that in terms of the number of complaints you will get, each of the options can be considered separately. To help us later, preprocess for each option, how many complaints we would get if we were to get the milk tea with that option and without it. We will now build up the best combinations one option at a time.

Let T_k denote the top 101 combinations that generate the fewest complaints when we consider only the first k options, represented as a binary string (tiebreaking arbitrarily). The key idea is to note that each combination in T_{k+1} has a combination from T_k as a prefix.

This is easy to show by contradiction. Take any string S from T_{k+1} and remove the last bit to get the prefix S'. Suppose for a contradiction that S' is not T_k. Taking any string in T_k and appending the removed bit will give a combination that generates strictly fewer (when considering the tiebreak) complaints. This gives 101 strings that generate fewer complaints, which contradicts S being in T_{k+1}.

So the final algorithm is as follows:

1. T_0 is the empty set.
2. To generate T_k (for k > 0), take each string in T_{k-1} and try appending both a 0 and a 1. This will give at most 202 potential answers, of which we keep the best 101 (in general, we keep the top **M**+1).
3. Take the best combination from $T_{\mathbf{P}}$ that isn't forbidden.

Naively, this can be done in $O(\mathbf{P}^2\mathbf{M})$, but can also be done faster in $O(\mathbf{PM})$. In total, the algorithm takes $O(\mathbf{PN} + \mathbf{P}^2\mathbf{M})$ or $O(\mathbf{PN} + \mathbf{PM})$.