Coding Competitions Farewell Rounds - Round A

Analysis: Untie

Test Set 1

In Test Set 1, the limits are small enough that we can try every possible string made entirely of `R`s, `P`s, and `S`s as the destination (there are only $3^{10} = 59049$ of them). For each one, we can check pairs of adjacent letters to make sure it is valid. For the valid destinations, we can see how many changes they require, and keep a running minimum of it.

This solution could be sped up by either generating the strings in a smarter way that does not generate lots of invalid cases only to filter them out later, or by precalculating all valid strings for each length only once, and then using that list directly for each test case. However, none of this is required to pass Test Set 1.

Test Set 2

For Test Set 2 we need a completely different approach, as there are way too many strings to try. We can observe that if we choose any one person, we can always make a choice for them that is different than their neighbors (because there are 2 neighbors and 3 possible choices). Therefore, any time we have two adjacent equal letters with their other neighbors different, we can always fix it with a single change, and we need at least one. Generalizing that, if we have exactly k consecutive letters that are equal, surrounded by different ones on both sides, we need at least $\lfloor k/2 \rfloor$ changes, and we can do it with exactly that many by changing alternating letters. Therefore, we can add that quantity for each run of consecutive letters (remembering to consider a prefix and a suffix that are made of the same letter as a single run) to the total.

There is one case that we need to solve differently, though. Notice that our precondition above is that a run of consecutive letters is surrounded by other letters. However, if all letters in the input are the same, there is no such run! This case is simple, though, as the answer is, as hinted by Sample Case #2, $\lceil$the length of $\mathbf{C}/2\rceil$.