

Analysis: Robot Programming Strategy

With A competitors, there are $A!$ possible initial setups for the tournament bracket, and A is itself exponential in the number of rounds of the tournament. A factorial of an exponential is terrifying enough, and we haven't even started the tournament yet!

Fortunately, we can ignore almost everything about the structure of the tournament. For any opponent, there is at least one possible initial setup in which we will play our first match against that opponent. Since we have to guarantee that we will win the tournament regardless of the initial setup, we must be able to defeat *every* opponent. We cannot tie an opponent, since that coin flip might not come up in our favor!

So all we have to do is find a program that beats every opponent's program. To check a program, we can just check it against each opponent, without worrying about the tournament setup.

Test set 1

In test set 1, there are at most 7 opponents, and their programs can be at most 5 characters long. Our program can be longer than that if need be, but how long does it need to be? We can observe that an optimal winning program should never waste a turn by tying with all remaining opponents, since then it could have instead chosen the move that would have beaten all of them; therefore, it should eliminate at least one opponent per move. So in test set 1, if there is a winning program, it is at most 7 moves long. We can comfortably check all $3^7 + 3^6 + \dots + 3$ possible programs of length up to 7.

When simulating a match, we cannot wait around for a googol of moves; by the same argument above, an optimal winning program should take no more than 7 moves to defeat all opponents, so we only need to simulate at most 7 moves. If we are still tied with the opponent at that point, we can safely give up on that program.

If we find that no program of length at most 7 beats every opponent's program, then we can declare that the case is IMPOSSIBLE.

Test set 2

In test set 2, there can be up to 255 opponents, and we cannot generate and check all programs of length up to 255. We must find a way to construct a winning program if it exists.

Let's imagine playing against all opponents at once. How do we choose our program's first move? We must win or tie against every opponent, so we will consider the set of their first moves. If it includes all three possible moves, we are doomed; no matter what we pick, at least one opponent will defeat us. If it includes only one move (e.g. every opponent starts with R), then we can pick the move that defeats that move (in this case, P) and instantly defeat everyone. Otherwise, the set includes two of the possible moves, and we should pick the move that ties one and beats the other. For example, if the opponents all lead off with S or P , we should pick S .

Eliminating any defeated opponents and proceeding to the next move of this combined "match", we can apply the same strategy, but considering the set of remaining opponents' second moves (looping through their programs as needed), and so on. We will eliminate at least one opponent with each move, so after A moves, we will either have our winning program or know that the

case is IMPOSSIBLE. Notice that this limit holds regardless of the lengths of the opponents' programs.