Towers

2021, Baraj 2, Seniori

1. We sort the towers and assume R_W <= R_G <= R_B.
2. If a tower is at least R_W away from both neighbours, we remove it (color it W).
3. Between two W, G/B can mostly alternate (at most one point where they don't).
   Since, if there are two dists >= R_W, the middle segment can freely have a W.
   IF there arem't, only one dist can be >= R_W and thus >= R_G (or even >= R_B).
4. There wont't (need to) be more than 4 non-W towers between two W:
   W G B G B G W -> R G B W B G W
   W G G B G B W -> W G W B G B W
   W B G G B G W -> W B G W B G W
   Other cases are symmetric.
5. We go through the towers in order keeping a DP: dp[pos][last]. We only store W
   positions, since those occur often enough in order to propagate data in O(1).
   Thus, `last` describes the last 2/3 towers and is one of: GW, BW, GWW, BWW.
   For GW and GWW the DP stores the leftmost possible position of the rightmost
   B tower before pos, such that there is a solution with that being the case.
   For BW and BWW -- the same but for a G tower. Notice that there can't be more
   than 2 W in a row because we removed such possible towers in 2.
6. Since we only need the rightmost positions of each color to propagate their
   effects, this gives us everything necessary to compute the DP in O(1) time per
   state (though with a big constant) by trying all possible options of length
   between 0 and 3 before the current GW/BW/GWW/BWW and trying each with each
   possible previous state. There are 20 potentially possible sequences * 4
   previous states = 80. This is also multiplied by 2 for each state we compute,
   (since the first color choice is already made), so a constant of 160 (or 80,
   if we only do it for GW, BW and then use that for the other two). If we do not
   take all constraints into account while exploring the possible sequences, we
   may try a few more than needed (30), which would give a higher constant. It
   should be possible to significantly reduce this constant by using an auxiliary
   DP, for segments of the sequence (4 * 2 = 8), so it might be a good idea to
   have a small subtask for reducing the constant factor of the solution.
7. Finally, to get an answer, we need to insert 1 fictitious towers in the end
   at dist = R_B from the last but without removing it in 2. Either the true last
   one is W or there is a solution where this one is. In either case, we can just
   check those two.