Problem

Determine the most efficient method to break the record in a speedrun. You may reset at any point.

Insights

```
During a run, you have r - n - 1 time margin to make errors.
```

Optimally, the only place where you reset is immediately after failing a trick.

Solution attempt

- Use dynamic programming!
- DP[i, j] := the expected time until a record when you are just before trick *i* and have used *j* margin for error. We are interested in DP[0, 0].

للعريد أفريح والمراجع فالمحافظ

- When you complete trick *i*, the rest of the run takes $(t_{i+1} t_i) + DP[i+1,j]$ time.
- When you fail the trick, you either reset (taking DP[0,0] time) or continue (taking $d_i + (t_{i+1} t_i) + DP[i+1, j+d_i]$ time).
- This gives a DP relation:

$$DP[i,j] = \begin{array}{cc} p_i & \cdot & ((t_{i+1} - t_i) + DP[i+1,j]) + \\ (1 - p_i) & \cdot & \min(DP[0,0], d_i + (t_{i+1} - t_i) + DP[i+1,j+d_i]) \end{array}$$

• We can use DP[m][j] = 0 as the base cases for the DP.

We now have a DP relation, but we need to know DP[0,0] in order to use it.

Solution

• Consider making some guess P for the value of DP[0,0]. We can use this value to fill the DP table.

للمعارف ويربين بتهريه كالمراجع

- When the resulting *DP*[0,0] is larger than *P*, the guess was too low. When *DP*[0,0] is smaller than *P*, the guess was too high.
- Use binary search to determine the optimal value of P, and thus the actual value of DP[0,0].

Statistics: 61 submissions, 8 + ? accepted