Ellen is a new Assembly Line Manager in a shoe factory.
So far, everything has been going very smoothly for her and N shoes of the same model and size have been
produced. Exactly half of them are left shoes and the other half are right shoes. The freshly sewn shoes
are standing in a line, in no particular order (i.e. with no regard to being left or right shoes).
They now need to be matched into pairs and packed into boxes. Ellen would like to assign this task to her
subordinate workers. Each worker should get a distinct interval of adjacent shoes, such that the number
of left shoes is equal to the number of right shoes. Each shoe must be assigned to exactly one worker.
What is the maximum number of workers that Ellen can assign to this task?

Write a function:

`fn find_max_par(s: String) -> usize`

that, given a string `s` of letters "L" and "R",
denoting the types of shoes in line (left or right), returns the maximum number of intervals such
that each interval contains an equal number of left and right shoes.

For example,
given `s` = "RLRRLLRLRRLL", the function should return 4, because `s` can be split into
intervals: "RL", "RRLL", "RL" and "RRLL". Note that the intervals do not have to be of the same size.

Given `s` = "RLLLRRRLLR", the function should return 4, because `s` can be split into
intervals: "RL", "LLRR", "RL" and "LR".

Given `s` = "LLRLRLRLRLRR", the function should return 1.
Write an efficient algorithm for the following assumptions:

N is an integer within the range [2..100,000];

string `s` consists only of the characters "R" and/or "L";
the number of letters "L" and "R" in string S is the same.

-1 +1 -1 -1 +1 +1 -1 +1 -1 -1 +1 +1

RLRRLLRLRRLL

-1 0 | -1 -2 -1 0 | -1 0 | -1 -2 -1 0   ④

RLLLRRRLLR

0 | 0 | 0 | 0   ④

LLRLRLRLRLRR

-1 -2 -1 -2 -1 -2 -1 -2 -1 -2 -1 -2 -1 0   ①