Kick Start 2022 - Round H

Analysis: Running in Circles

Test Set 1

Since Ada only runs in the clockwise direction, we can ignore the direction input altogether as Ada will always cross the starting line in the same direction she had last crossed it. Now, we can calculate the total distance run by Ada by calculating the sum:

$$dist = (\mathbf{D_1} + \mathbf{D_2} + \cdots + \mathbf{D_N})$$

This sum can be calculated while reading the input line-by-line and adding to $dist$, so we do not need to store the value of each $\mathbf{D_i}$. Then, we can calculate $\lfloor \frac{dist}{\mathbf{L}} \rfloor$, and obtain the number of laps the machine has counted.

$$laps = \lfloor \frac{dist}{\mathbf{L}} \rfloor$$

Time and Space Complexity: This solution will run in $O(\mathbf{N})$ time and take up $O(1)$ extra space.

Test Set 2

Going through the input line-by-line, we calculate how much Ada has to run to be able to reach the starting line again (that is, the remaining distance to starting line, R) from the Ada's current position pos ($pos = 0$ initially) before a run and in the current direction $\mathbf{C_i}$. We also keep track of the previous direction $prevdir$ she was running in when she last crossed the starting line.

R can be calculated as follows:

- If $\mathbf{C_i}$ is clockwise, $R = (\mathbf{L} - pos) \bmod \mathbf{L}$
- If $\mathbf{C_i}$ is anticlockwise, $R = pos$

During each run (that is, for each line of input):

- If $\mathbf{D_i} < R$, then we will just update her position pos, as she can not reach the starting line.
  - If $\mathbf{C_i}$ is clockwise, $pos = pos + \mathbf{D_i}$
  - If $\mathbf{C_i}$ is anticlockwise, $pos = pos - \mathbf{D_i}$

  After this update, we continue to the next run without updating $prevdir$ as Ada has not crossed the line this time.

- On the other hand if $\mathbf{D_i} \geq R$, and if $R > 0$, we increase the number of laps by 1 if $\mathbf{C_i}$ is the same as $prevdir$. However, if $R = 0$ (that is, Ada is exactly at the starting line), we do not perform this increment since Ada had already reached the starting line in the previous run, where it was counted as a lap by the machine (similarly for $R < 0$). Thus, we do not treat starting a run at the line as "crossing" the line, rather, we count a lap when Ada finishes at the line. With this logic, the case where Ada changes her direction on the line itself will also not be counted as a lap.

  We then increase the number of laps by the value $\lfloor \frac{\mathbf{D_i} - R}{\mathbf{L}} \rfloor$ to take care of cases where multiple laps are made in a single run (this will also handle the above case, where Ada just reaches the starting line after a run). As these laps are counted from the starting line, $prevdir$ does not matter in this calculation. Then, we update Ada's current position:

- If $\mathbf{C_i}$ is clockwise, $pos = (\mathbf{D_i} - R) \bmod \mathbf{L}$
- If $\mathbf{C_i}$ is anticlockwise, $pos = \mathbf{L} - ((\mathbf{D_i} - R) \bmod \mathbf{L})$

After the above process, we update $prevdir$ to $\mathbf{C_i}$ and continue.

Finally, return the number of laps counted.

Time and Space Complexity: This solution also runs in $O(\mathbf{N})$ time, and takes up $O(1)$ extra space.