Dishonest Driver

- There is probably multiple solutions, but this is mine

- C is a compressed path if
  1. C is a letter (A-Z) or number
  2. $C = A+B$ where A & B are paths concatenated
  3. $C = A^n$ where A is a compressed path repeated n times

- From this we can easily setup a recurrence
  DP[i][j]: is the size of the smallest compressed path creating substring C[i...j]

$$DP[i][j] = \begin{cases} 1 & \text{if } i=j \quad \text{Base case (1)} \\ \min_{i \le k < j} DP[i][k] + DP[k+1][j] & \text{Concatenation (2)} \\ \min_{1 \le k \le j-i+1} DP[i][k] & \text{if } C[i \ldots i+k-1] = C[i+k \ldots i+2k-1] = \ldots = C[i+N\cdot k \ldots j] \quad \text{(3) repeating substring} \end{cases}$$

- Now we have two questions
  1. How can we efficiently check for repeating substrings
  2. What is the complexity of our recurrence

Q1. We can compare substrings in $O(1)$ with some preprocessing and hash functions
For a string A $H(i) = \sum_{i=0}^{n} A_i \cdot p^i$ so a prefix sum using a hash function

if we want $H(i,j) = [H(j) - H(i-1)]\frac{1}{p^i}$ so to check if $A[i \ldots j] = A[k \ldots L]$ just use $H(i,j) = H(k,L)$ $O(1)$!!

Now with this compute $A[i][j][k] = 1$ if we can repeat $A[i \ldots j-1]$ k times starting at index i
0 otherwise

e.g. aaab $A[0][1][2] = 1$ since we have aaab
$A[1][1][1] = 1$ since we have aaab

so total $\sim O(n^3)$

Q2. Part 3 of our dP is just $O(n^3)$ easy to show
Part 2 is a bit more tricky: But if we fix a length and starting point we can just iterate over the divisors of the length and check if we can repeat that substring to get $C[i \ldots j]$ in $O(1)$
so we get $O(n^2 \times \sqrt{n})$
in total our complexity is $O(n^3)$