



3

JOI エリミネーター (JOI Eliminator)

Author : 米山 瑛士

小課題 1

S の中で J, O, I がこの順で連続して並んでいる箇所を探し、その部分を O, I, J の並びに置き換えることを繰り返せば良い。

操作の回数は $O(N^2)$ である。

証明

J のある場所のインデックスの総和は、はじめの状態で $O(N^2)$ であり、操作の度に 2 減少する。この値は操作の終了時 0 以上であるため、操作の回数は $O(N^2)$ 回である。

小課題 2

操作は、 OI の並びと J を入れ替える操作だと考えることができる。

答えは、 $OIOI \dots OIOIJJ \dots JJ$ のように、 OI を $N/3$ 回繰り返し、続いて J を $N/3$ 回繰り返した文字列である。

証明

まず、以下の「不変条件」を示す。

- S に現れる J の個数は操作によって変わらない。
- S に現れる OI の並びの個数は操作によって変わらない。

一つ目は明らかである。二つ目は、操作をする三文字のインデックスを $i, i+1, i+2$ としたときに、 $S[i-1:i]$ と $S[i+2:i+3]$ がともに、操作の前と後のいずれでも OI とならないことから確認できる。

次に、文字列 $JOIJOI \dots JOI$ についての「初期条件」を確認する。 J の個数および OI の個数は $N/3$ である。

したがって、最終的な文字列においても J の個数および OI の個数は $N/3$ である。



最終的な文字列は、以下の条件を満たす。

- JOI を含まない。
- J を $N/3$ 個と OI を $N/3$ 個を並べた文字列である。

このような条件を満たす文字列は、 $OIOI\dots OIOIJJ\dots JJ$ のみである。

小課題 3

答えは、 S の $k+1$ 文字目以降で OI が連続する箇所を切り取って、先頭に移動させたものである。

満点解法

はじめの状態で OI の並びに含まれないような O や I は、今後一切 JOI の並びに含まれることが無いため、今後一切操作を受けることがなく、固定されたものだと考えてよい。

そのような文字で文字列を切り分けて考えると、それぞれの部分は OI と J が並んだ文字列となっている。

このような部分文字列は、小課題 2 と同様に、最終的に OI が前にすべて並び、J が後にすべて並んだ文字列に変化する。

実装の際には、文字列を前から見て、OI を見つけたらすぐ出力し、一方で J を見つけたらその個数だけカウントしてスルーしておき、固定された O や I を見つけるか文字列の末尾に到達したときにまとめて出力するという処理を行えばよい。計算量は $O(N)$ である。