

JOI 2011/2012 本選

01: JJOOII

解説：保坂 和宏

問題概要

- J, O, I からなる文字列 S が与えられたとき, S に含まれる JOI 列のレベルの最大値を求める

「レベル 0 の JOI 列」 … (空文字列)

「レベル 1 の JOI 列」 … JOI

「レベル 2 の JOI 列」 … JJOOII

「レベル 3 の JOI 列」 … JJJOOOIII

.....

問題概要

- 例 : $S = \text{OJJOOIIOJOI}$
 - レベル 0 の JOI 列は含まれる
 - レベル 1 の JOI 列は含まれる (OJJOOIIOJOI)
 - レベル 2 の JOI 列は含まれる (OJJOOIIOJOI)
 - レベル 3 以上の JOI 列は含まれない
- JOI 列はつながって現れるものを考える
 - JOIJOIJOIJOIJOI にはレベル 2 の JOI 列は含まれない

問題概要

- 制約：文字列 S の長さ N は, $1 \leq N \leq 10^6$
 - 部分点 (20%) : $1 \leq N \leq 100$

単純な解法

- 各 k に対し, レベル k の JOI 列が S に含まれるかどうかを調べる
 - $0 \leq k \leq N/3$ を考えればよい
 - S の中でどこから始まるかをすべて考える
 - $3k$ 文字がレベル k の JOI 列であるか調べる
 - 時間計算量 $O(N^3)$

やや効率の良い解法

- 各 k に対し, レベル k の JOI 列が S に含まれるかどうかを調べる
 - $0 \leq k \leq N/3$ を考えればよい
 - S の中でどこから始まるかをすべて考える
 - $3k$ 文字がレベル k の JOI 列であるか調べる
 - 時間計算量 $O(N^3) \rightarrow O(N^2)$
- レベル k の JOI 列であるかの判定を高速にしたい

やや効率の良い解法

- 「 S の $i + 1$ 文字目から $i + k$ 文字目までがすべて文字 x であるか」がわかればよい
 - ($i + 1$ 文字目から $i + k$ 文字目までの x の個数)
= ($i + k$ 文字目までの x の個数)
- (i 文字目までの x の個数)
 - 「 S の j 文字目までの文字 x の個数」を各 j ($0 \leq j \leq N$) について予め計算しておけばよい
 - 前処理 $O(N)$ / 判定 $O(1)$

やや効率の良い解法

J J O J I O J J J O O I I
0 1 2 2 3 3 3 4 5 6 6 6 6 6

J J O J I O J J J O O I I
0 1 2 2 3 3 3 4 5 6 6 6 6 6

J J O J I O J J J O O I I
0 1 2 2 3 3 3 4 5 6 6 6 6 6

効率の良い解法 (1)

- レベル k の JOI 列 ($k \geq 1$) の特徴的な部分
 - 文字 O が k 個連続している
 - その左右には O でない文字が来ている

JJJJOOOOIIII

効率の良い解法 (1)

- S に「文字 O が k 個連続していて、その左右には O でない文字が来ている」というような箇所があれば、その部分を使う JOI 列のレベルは k

× ...JOOOO|...

JJJOOOIII

× JJJJOOOOIIII

JJJJJOOOOOIIII

効率の良い解法 (1)

- S の「文字 0 が連続していて、その左右には 0 でない文字が来ている」というような箇所をすべて求める
 - $O(N)$
- それぞれの箇所に対し、文字 0 の個数を k としたとき、左側 k 文字がすべて J であるか・右側 k 文字がすべて I であるかを判定する
 - 先程の方法を用いれば、各箇所について $O(1)$

効率の良い解法 (1')

- それぞれの箇所に対し、文字 0 の個数を k としたとき、左側 k 文字がすべて J であるか・右側 k 文字がすべて I であるかを判定する
 - 実は工夫は不要で、左側 k 文字・右側 k 文字をすべて見てもよい
 - (見ることになる文字数) = ($2k$ の和)
= $2 \times (S$ に現れる文字 0 の個数) $\leq 2N$

効率の良い解法 (2)

- JOI 列を探しやすくするように S を加工
- Run-Length Encoding
 - 文字列の圧縮に用いられる基本的な手法の 1 つ
- 同じ文字が連続する箇所をその長さで表す

OOJJJJJJOOOOIIII



[O × 2] [J × 7] [O × 4] [I × 5]

効率の良い解法 (2)

- Run-Length Encoding を行った結果に対して, JOI 列は $[J \times a] [O \times b] [I \times c]$ という形の箇所にはしか現れない
 - $a \geq b$ かつ $b \leq c$ のとき, レベル b の JOI 列が現れる
 - そうでないとき, JOI 列は現れない
 - 例 :
 - $(a, b, c) = (7, 4, 5)$: JJJJJJJOOOOIIII
 - $(a, b, c) = (1, 2, 3)$: JOOIII

効率の良い解法 (2)

- Run-Length Encoding を行う
 - 時間計算量・空間計算量 $O(N)$
- 連続する 3 パートごとに,
[J × a] [O × b] [I × c] ($a \geq b$ かつ $b \leq c$) と
いう形かを確認する
 - $O(N)$

効率の良い解法 (2')

- 簡単に言い換えたアルゴリズム (メモリも節約)
- 以下を繰り返す：
 - S の先頭から J, O, I 以外の文字を取れるだけ取る (今回は不要)
 - S の先頭から文字 J を取れるだけ取る
 - S の先頭から文字 O を取れるだけ取る
 - S の先頭から文字 I を取れるだけ取る
 - J を a 個, O を b 個, I を c 個取れたとして, $a \geq b$ かつ $b \leq c$ ならばレベル b の JOI 列あり

まとめ

- JOI 列を効率良く探す
- JOI 列をどこから探せばよいかを考える
- Run-Length Encoding
- その他の解法もいろいろ

結果

