

# Binary Strings 解説

原案 / writer / 解説: Forested

tester: shiomusubi496

## 問題概要

長さ  $N$  の 01 文字列  $S$  であって、 $i = 1, 2, \dots, M$  について次の条件を満たすものを数え上げよ。

- $L_i$  文字目から  $R_i$  文字目に文字  $C_i$  が含まれる

## 小課題

1. (50 pts)  $N \leq 16, M \leq 16$
2. (100 pts)  $N \leq 2000, M \leq 2000, C_i = 0$
3. (150 pts)  $C_i = 0$
4. (300 pts) 追加の制約はない

## 小課題 1

$N, M$  が非常に小さい  $\rightarrow$  全探索しても間に合う

愚直に判定しても  $O(2^N NM)$  時間で間に合う

累積和で判定を高速化すると  $O(2^N (N + M))$  時間になる

## 小課題 2

$N, M$  が 2000、 $C_i$  が 0

とりあえず多項式時間に見してみよう

## 小課題 2

こういうことが起きてはいけない

1	0	1	1	1	0
---	---	---	---	---	---

ここに0

## 小課題 2

$k$  文字目まで決めたら、 $R_i = k$  となる  $i$  について  $(L_i, R_i)$  の条件を満たすか確認することにする

1	0	1	0	1	
---	---	---	---	---	--

ここに 0

ここに 0

## 小課題 2

$k$  文字目まで決めたら、 $R_i = k$  となる  $i$  について  $(L_i, R_i)$  の条件を満たすか確認することにする

1	0	1	0	1	
---	---	---	---	---	--

ここに 0

ここに 0

どうやって？

## 小課題 2

$k$  文字目まで決めたら、 $R_i = k$  となる  $i$  について  $(L_i, R_i)$  の条件を満たすか確認することにする

1	0	1	0	1	
---	---	---	---	---	--

ここに 0

ここに 0

どうやって？ → 一番右の 0 の位置が分かれば良い

## 小課題 2

$DP[i][j]$  :  $i$  文字目まで決めて、一番右にある 0 が  $j$  文字目であるような文字列の数 (0 が存在しない場合は  $DP[i][0]$  として数える)

$R_k = i$  となる  $k$  について、 $DP[i][0], DP[i][1], \dots, DP[i][L_k - 1]$  を 0 にする

$O(N^2 + M)$  時間

## 小課題 3

$C_i$  が 0、 $N$  と  $M$  には制限がない

小課題 2 を高速化しよう

## 小課題 3

$R_k = i$  となる  $k$  について、 $DP[i][0], DP[i][1], \dots, DP[i][L_k - 1]$   
を 0 にする

高速化できそう？

## 小課題 3

DP の遷移をちゃんと書いてみる

- $DP[i][i] = DP[i-1][0] + DP[i-1][1] + \dots + DP[i-1][i-1]$
- $DP[i][j] = DP[i-1][j] (j < i)$
- $DP[i][0], DP[i][1], \dots, DP[i][t-1]$  を 0 にする

## 小課題 3

DP の遷移をちゃんと書いてみる

- $DP[i][i] = DP[i-1][0] + DP[i-1][1] + \dots + DP[i-1][i-1]$
- $DP[i][j] = DP[i-1][j] (j < i)$
- $DP[i][0], DP[i][1], \dots, DP[i][t-1]$  を 0 にする

→in-place に更新できそう

## 小課題 3

- ある要素を更新する
- ある区間に 0 を代入
- ある区間の和を求める

例えば遅延セグ木でできる

$O((N + M) \log N)$  時間

## 小課題 4

DP の情報を増やす

- $DP[i][j][k]$  :  $i$  文字目まで決めて、一番右の文字は  $j$  で、 $j$  でない方の文字のうち一番右にあるのは  $k$  文字目であるような文字列の数

同じように遅延セグ木で in-place に更新できる

$O((N + M) \log N)$  時間

# おまけ

- $O(N + M)$  時間で解けます
  - 整理するとやるべきことが prefix への 0 の代入と全体の和の取得だけになるので、queue 等のできる
- 包除原理を用いても解けます

全体 FA: PCTprobability (24:01)

オンサイト FA: 同上