

J: Forbidden Word

原案: harurun

問題文: harurun

データセット: not

解答: harurun, hos, kuma, not, riantkb

解説: not

問題概要

- 文字列SとTが与えられる。
- 文字列Sをある幅の複数行で表示した時、縦か横のどこかでTが出現するか、それぞれの幅について判定せよ。

- $|S| = N, |T| = M$
 - $1 \leq M \leq N \leq 200,000$
- (例) S: banana, T: ana

b	ba	ban	bana	banan	banana
a	na	ana	na	a	
n	na				
a					
n					
a					

解法:横方向

1. 改行を気にせずSにTが出現する時の先頭の位置を求める
2. ある位置について、次にTが出現する位置を求める
3. 幅を順番に試し、各行で最初のTの出現位置が改行を跨がないか調べる

[例] S=banana, T=ana

1.	banana	b	ba	ban	bana	bana	bana
2.	113366	a	na	<u>ana</u>	na	a	
		n	na				
		a					
		n					
		a					

解法:横方向

1. 改行を気にせずSにTが出現する時の先頭の位置を求める
KMPなどで $O(N)$ またはbitsetで $O(ML/w)$ (L: bitsetの長さ・w: word size)
2. ある位置について、次にTが出現する位置を求める
後ろから見ていくと $O(N)$
3. 幅を順番に試し、各行で最初のTの出現位置が改行を跨がないか調べる
行数の総和: $N + N/2 + N/3 + \dots \rightarrow O(N \log N)$

解法：縦方向

1. Tについて1文字目から順にSにその文字が出現する場所を求める
2. 文字数分ずらし、全ての文字について出現位置が重なるか調べる

[例] S=banana, T=ana

b a n b a n a n a
a n a n a n a
n a n n a
a n a
n a
a

解法:縦方向

1. Tについて1文字目から順にSにその文字が出現する場所を求める
文字ごとにSでの出現位置を先に計算しておく $O(N)$
2. 文字数分ずらし、全ての文字について出現位置が重なるか調べる
幅のパターンが $O(N/M)$
それぞれの幅でずらす回数が $O(M)$
1回ごとの重なるか調べるのはbitsetで処理すると $O(L/w)$
→ 全体の計算量は $O(NL/w)$

ジャッジ解

● harurun	C++	112 lines,	2.5kB
● hos	C++	110 lines,	3.0kB
● kuma	C++	219 lines,	6.4kB
● not	C++	67 lines,	1.2kB
● not (astra)	C++	56 lines,	1.6kB
● not (astra)	Python	54 lines,	1.5kB
● riantkb	Python	29 lines,	843B

統計情報

- AC teams
 - 0 + 0 teams
- First Acceptance
 - N/A