

ACM-ICPC Japan Alumni Group

模擬地区予選 2015

Problem A : M and A

原案 : 山口

問題文 : 山崎

データセット : 矢藤

解答 : 澤, 矢藤, 山口, 山崎

解説 : 山崎

問題概要

- 文字列 S, T が与えられる
- S, T から文字を「交互にとって」 S を作れるか？
 - S, T の部分列をそれぞれ s, t とする
 - 2つの条件
 - $s_0 + t_0 + s_1 + t_1 + \dots = S$
 - $t_0 + s_0 + t_1 + s_1 + \dots = S$
 - の内少なくとも1つは成り立つ s, t は存在するか？
- $|S| = |T| \leq 1,000$

考察

逆向きに考える

- 「 s, t から S をつくる」ではなく「 S から s, t をつくる」
- 作った s, t がそれぞれ S, T の部分列になっていればよい

- 例 : $S = \text{"abcdefg"}$ のとき
 - 奇数文字目をとった文字列 = "aceg"
 - 偶数文字目をとった文字列 = "bdf"
 - この 2 つにそれぞれ一致する s, t をとればよい

考察

S 側は考えなくてよい

- S の奇数 or 偶数文字目を抜き出したら明らかに S の部分列
- s の候補は必ず条件を満たす
- t についてだけ調べればよい

t のとり方は 2 通り

- 奇数番目にするか偶数番目にするか
- この内少なくとも一方が T の部分列ならば "Yes"

解法

1. S の奇数文字目, 偶数文字目を取り出した文字列をつくる
 - 書き込み先文字列を 2 つ作って振り分けていく
 - $(i \% 2 ? s1 : s0) += S[i];$
 - $O(|S|)$
2. 少なくとも一方が T の部分列になっているかを調べる
 - $O(|T|)$
 - 詳細は次ページ

部分列の判定

- t の先頭から順に, T の一致する文字にマッチさせていく
- 文字が一致する位置が複数あるとき, より手前の位置を採用
 - 後ろを採用しても, 以降使える文字(選択肢)が減るだけで損

```
bool subseq( const string &T, const string &t )
{
    size_t pos = 0;
    for ( const char c : t )
    {
        pos = T.find( c, pos );
        if ( pos == string::npos ) return false;
        ++pos;
    }
    return true;
}
```

ジャッジ解

- 澤
 - 26 lines / 580 bytes (C++)
- 矢藤
 - 20 lines / 413 bytes (C++)
- 山口
 - 20 lines / 596 bytes (Java)
- 山崎
 - 40 lines / 640 bytes (C++)

それぞれコメント込み

結果

- AC / Submissions
 - 55 / 139 (40 %)
- AC / Trying
 - 55 / 59 (93 %)
- First Acceptance
 - anta (03:55)