

立命合宿 2017 Day3

A問題「アルファベットブロック」

原案 : 杉江

問題文 : 杉江

テスター : 杉江、鈴木、井上、田中

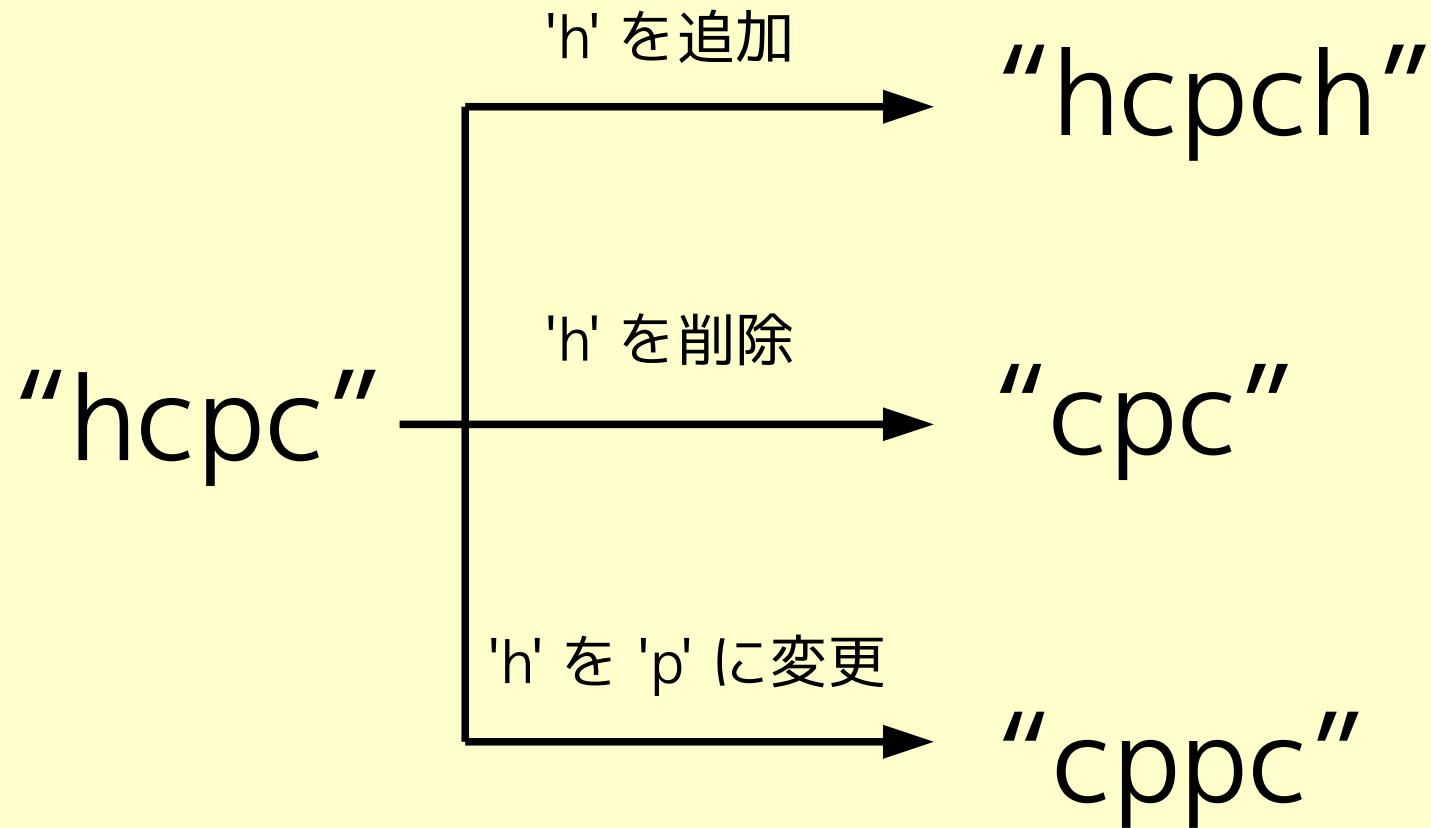
解説 : 杉江

問題概要

- 英小文字のみからなる文字列 S が与えられる
 - $1 \leq |S| \leq 10^3$
- S に対して以下の操作を何度か行って、文字を並び替えたなら回文ができあがるようにしたい
 - ✓ ある 1 文字を S に追加
 - ✓ S 中のある 1 文字を削除
 - ✓ S 中のある 1 文字を別の文字に置き換える
- 回文にするための最小操作回数を求めよ

サンプル

例: "hcpc"



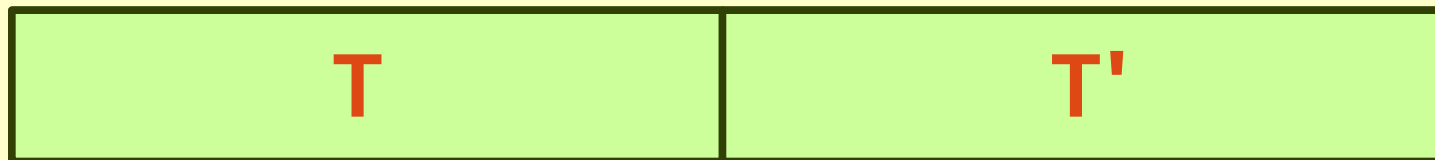
- コストはすべて 1

考察 1: 回文の性質

- 回文とは、右から読んでも左から読んでも同一の文字列が得られる文のことである
- S の長さを N とおき、文字列の前半 ($N/2$) 文字からなる部分文字列を T 、 T を左右反転させた文字列を T' とする
- $S \rightarrow T + (\text{真ん中の文字}) + T'$ が成り立つ



(N が奇数)



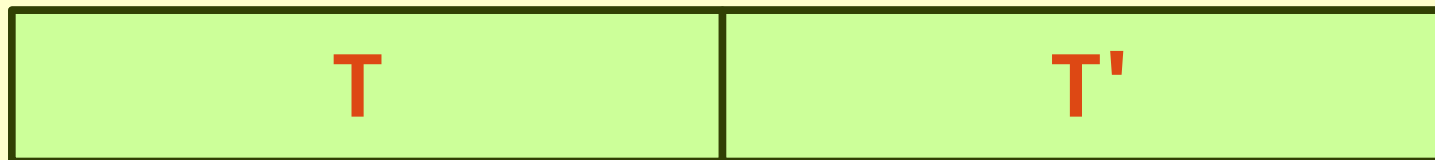
(N が偶数)

考察 2: 回文の性質

- T' は T を左右反転させたものなので、登場する文字の種類とその個数は T と等しい
- つまり $T + T'$ では、全てのアルファベットが偶数個ずつ存在する



(N が奇数)



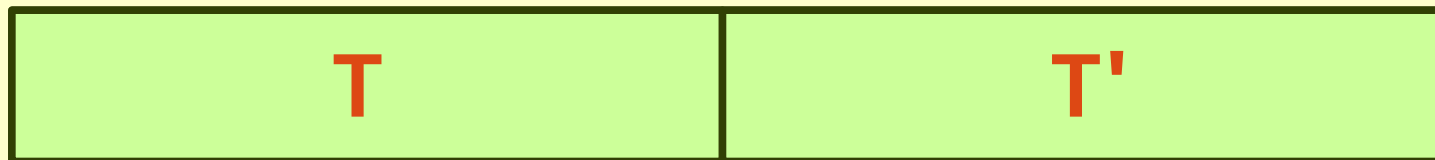
(N が偶数)

考察 3: 回文の性質

- N が奇数のとき
 - 1 種類だけが奇数個、他の種類のアルファベットが偶数個ずつ存在すれば回文になる
- N が偶数のとき
 - 全てのアルファベットが偶数個ずつ存在すれば回文になる



(N が奇数)



(N が偶数)

考察 4: 解法

- 最初から偶数個あるアルファベットに関しては、それ以上操作の必要がない
- 奇数個あるアルファベットを偶数個にするための最適な操作はなにか？
- 今回の場合、**「S 中のある 1 文字を別の文字に置き換える」**操作があるのでそれを使おう

考察 5: 解法

- 奇数個ある文字を、別の奇数個ある文字に置き換えるのが最適
- 文字 C_1 が $(2p+1)$ 個、文字 C_2 が $(2q+1)$ 個あったとして、 C_1 を C_2 に置き換える
 - ✓ C_1 は $(2p+1) - 1 = 2p$ 個 (偶数)
 - ✓ C_2 は $(2q+1) + 1 = 2(q+1)$ 個 (偶数)
- 一度の操作で 2 種類のアルファベットが偶数になる！

解法まとめ

- 奇数個ある文字の種類を数える (M とおく)
- 答えは、 $M / 2$ (切り捨て) になる
- (追加や削除の操作は実は使いません)

ちなみに

- もし「置き換えの操作」がない場合は、 $\max(0, M - 1)$ が答えになります
- 先ほどと同様に考えればこれが導けます
(各自やってみましょう)

Tester 解

- 杉江 : 241 bytes, 10 lines (C++)
- 鈴木 : 293 bytes, 15 lines (C++)
- 井上 : 237 bytes, 16 lines (C++)
- 田中 : 422 bytes, 13 lines (Java)
- 田中 : 125 bytes, 12 lines (Python3)

統計

- **Accept / Submission**
 - 58.54 % (24/41)
- **First Acceptance**
 - On-site : Gachofriends (11 min)
 - On-line : satanic0258 (4 min)