ICPC模擬国内予選2021
D: ぶりしゃぶしゃぶり部

原案: darsein
問題文: amylase
データセット: not
解答: amylase, cos, not, riantkb
解説: climpet

問題概要

- 文字列 S が与えられる。
- S の後半 (奇数文字の場合は中央の文字を含まない) から 0 個以上の区間を重ならないように選び、それぞれの区間の部分文字列を反転させる。それぞれの区間につき、区間長の 2 乗のコストがかかる。
- 区間を適切に選ぶことで、文字列を回文にできるか判定せよ。回文にできるならば、そのために必要なコストの最小値を求めよ。

- 制約:
  - $1 \leq |S| \leq 20{,}000$

考察

- S の前半を A、後半の逆文字列を B とする。
- すなわち、$n = \lfloor |S| / 2 \rfloor$, $A = S[1..n]$, $B = \mathrm{rev}(S[|S|-n+1..|S|])$ とする。ただし $\mathrm{rev}(x)$ は x の逆文字列。
- 区間 $[f..t]$ が次の条件を満たすとき、その区間を「反転可能区間」と呼ぶことにする。
  - $A[f..t] = \mathrm{rev}(B[f..t])$
- 元の問題は、次のように言い換えられる。
  - 反転可能区間からいくつかの区間を重ならないように選び、それぞれの区間に対応する B の部分文字列を反転させる。B を A に一致させるためのコストを求めよ。

$O(|S|^2)$ 時間解法

- 次のような単純な動的計画法が考えられる。
  a. dp[i] = (B[1..i] を A[1..i] に一致させるためのコスト) とする。初期値は dp[0] = 0、それ以外は∞。
  b. for f in 1..n
     - A[f] = B[f] ならば、dp[f] ← min(dp[f], dp[f−1])
     - for t in t..n
       - [f..t] が反転可能区間であるならば、dp[t] ← min(dp[t], dp[f−1] + $(t-f+1)^2$)
  c. dp[n] が答え。dp[n] = ∞ の場合、−1。
- f で始まる反転可能区間の列挙は、Z algorithm やローリングハッシュなどを用いることで、効率的に求められる。
- 計算量は全体で $O(|S|^2)$。
  a. 実行には数十秒程度かかることがあるが、国内予選の形式であればなんとかなる。

貪欲法

- 実は、ある位置で始まる反転区間は、最短のもののみ考えればよい。すなわち、次のような貪欲法でも正しい答えが得られる。
  a. $f \leftarrow 1$
  b. while $f \leq n$
     - $A[f] = B[f]$ ならば、$f \leftarrow f + 1$。すなわち反転を行わない。
     - そうでない場合、f から始まる反転区間のうち **最短のもの** $[f..t]$ を見つけ、この区間を反転、すなわちつまり答えに $(t-f+1)^2$ を加算する。さらに $f \leftarrow t + 1$ とする。
       - ただし、f で始まる反転区間が存在しなければ、答えは -1。
- なぜ？

貪欲法の正当性の証明

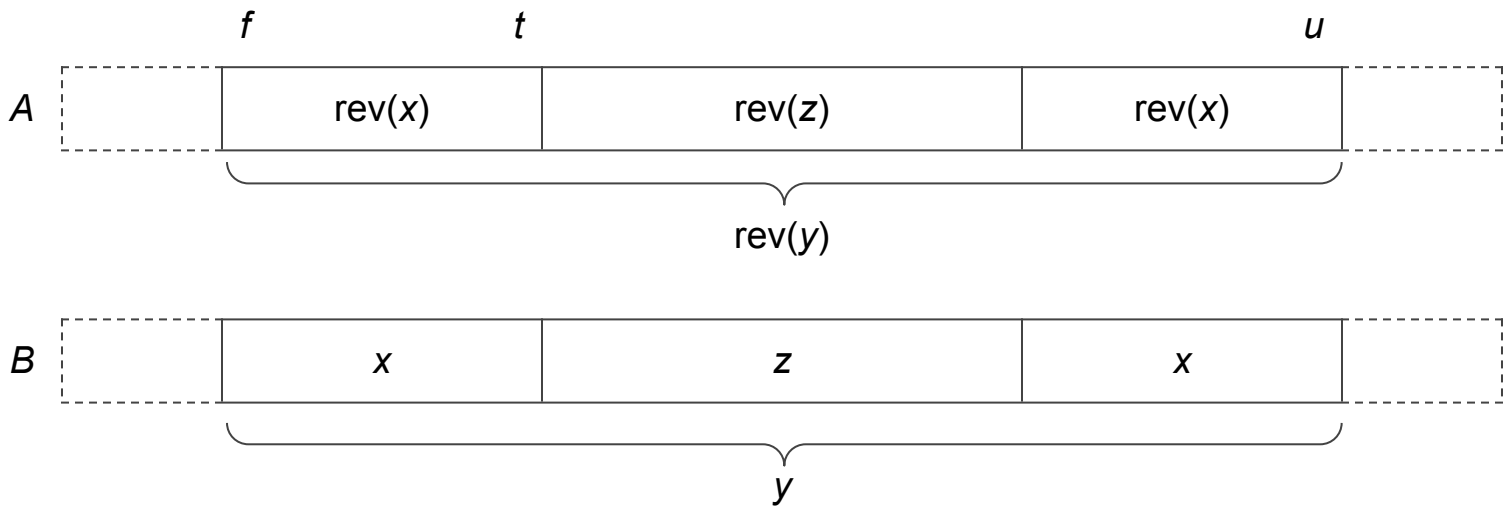
- f を固定する。まずは $A[f] = B[f]$ の場合を考える。f で始まる任意の反転可能区間 $[f..u]$ について、その区間を反転させる必要がないことを示す。
    - $f = t$ の場合、明らかに無駄。
    - それ以外の場合でも、$A[f], A[u], B[f], B[u]$ はすべて同じ文字となるため、$[f..u]$ の代わりに $[f+1..u-1]$ を反転させてもよく。こちらの方が得である。
  - いずれの場合も $[f..u]$ を反転させる必要はない。

貪欲法の正当性の証明

- 次に $A[f] \neq B[f]$ の場合を考える。f で始まる反転可能区間が複数存在し、そのうち最短のものが $[f..t]$、それ以外のものが $[f..u]$ であったとする。このとき、$[f..u]$ を反転させる必要がないことを示す。
- $x = B[f..t], y = B[f..u]$ とおく。x は y の接頭辞かつ接尾辞となる。
  - 証明：接頭辞は明らか。接尾辞は、$[f..u]$ と $[f..t]$ がともに反転可能区間であることから、$y[|y|-|x|+1..|y|] = B[u-t+f+1..u] = \mathrm{rev}(A[f..t]) = B[f..t] = x$。
  - このように、ある文字列の接頭辞かつ接尾辞であるような文字列を、その文字列のボーダーという。
- $|y| \geq 2|x|$ かで場合分けを行う。

貪欲法の正当性の証明

- $|y| \geq 2|x|$ の場合、y はある文字列 z を用いて $x + z + x$ と書ける。$[f..u]$ の代わりにこれら 3 つの文字列を別々に反転させても同じ結果が得られ、その方が得。



貪欲法の正当性の証明

- $|y| < 2|x|$ の場合を考える。
- x が y のボーダーであることから、$d = u - t$ $(= |y| - |x|)$ とおくと、x は周期 d を持つ。すなわち、任意の i に対し $x[i] = x[i + d]$ となる。
  - つまり、x は “abcabcabcab” のような文字列である。
  - 詳しくは このあたり を参照。
- ここから、$[f..t-d]$ もまた反転可能区間であることが言える。
  - $|y| < 2|x|$ の仮定から $f \leq t - d$、すなわちこの区間が空でないことが言える。
- これは、$[f..t]$ が f で始まる**最短の**反転可能区間であるという仮定に矛盾する。
- よって、$|y| < 2|x|$ の場合を考慮する必要はない。

貪欲法

- 以上の考察から、貪欲法が正しい答えを返すことがわかる。
- あとは、ある位置 f で始まる反転可能区間のうち最短のもの $[f..t]$ が求められればよい。これは、短い区間から順に試していくだけでよい。
- 接尾辞配列やローリングハッシュなどの手法により部分文字列の比較を高速に行えば、全体で $O(|S|)$ 時間を達成できる。

統計情報

- AC / trying teams
  - 88 / 93
- First acceptance
  - tonosama (20:38)