ICPC模擬国内予選2019
D: 文字列の魔法

原案: climpet
問題文: climpet
データセット: not
解答: climpet, not, smiken, hos
解説: climpet

問題概要

文字列 X に以下の操作を繰り返して別の文字列 Y にする。コストの最小値を求めよ。

- 挿入 (コスト A): 1文字挿入
- 削除 (コスト E): 1文字削除
- 置換 (コスト S): 1文字を別の文字に置換
- 回転 (コスト R): 先頭の1文字を末尾に移動

制約

- $|X|, |Y| \leq 100$

回転がない場合

挿入・削除・置換だけなら典型問題であり、以下の2次元DPで計算できる。
$dp(|X|, |Y|)$ が最終的な編集コスト。

- $dp(0, 0) = 0$
- $dp(i, j) = \infty$ ($i < 0$ または $j < 0$ のとき)
- $dp(i, j) = \min\{dp(i, j-1) + A, dp(i-1, j) + E, dp(i-1, j-1) + \delta(i, j)\}$
  ($0 \le i \le |X|, 0 \le j \le |Y|, i > 0$ または $j > 0$ のとき)
  ただし $X[i-1]=Y[j-1]$ のとき $\delta(i, j) = 0$, それ以外のとき $\delta(i, j) = S$

なんだかよくわからないという人は、「編集距離」または「レーベンシュタイン距離」で検索

- 今回の問題は対称性を満たさないため「距離」ではないが、求め方は同じ

回転がある場合 (想定誤解法)

次の方法を考えてみる

1. ある整数 k を1つ決める
2. 「X を k 回回転させた文字列」から Y への編集コストを求める
3. 上記の編集コストに回転コスト kR を足したものを答えの候補とする

すべての k $(0 \leq k < |X|)$ についてこれらを試せば、その最小値が答えとなる…？

このままでは実はだめ

考察

例えば、X = abc, Y = cb とすると

- abc --a 回転、コスト R--> bca --b 回転、コスト R--> cab --a 削除、コスト E--> cb
- abc --a 削除、コスト R--> bc --b 回転、コスト R--> cb

では、後者の方が明らかにコストが小さい

最終的に a を削除することを考えると、前者における「a 回転、コスト R」は無駄

回転された文字を削除する場合、支払い済みの回転コスト R を打ち消せば辻褄は合う

- つまり abc --a 回転、コスト R--> bca --b 回転、コスト R--> cab --a 削除、コスト $\boldsymbol{E-R}$---> cb

想定解法

以上の考察から、次の解法が得られる

1. ある整数 k を1つ決める
2. 「X を k 回回転させた文字列」から Y への編集コストを求める。
   - ただし、削除コストについては、
     前 $|X|-k$ 文字分は E のまま、後ろ k 文字分は $E-R$ とみなす。
3. 上記の編集コストに回転コスト kR を足したものを答えの候補とする

すべての k $(0 \leq k < |X|)$ についてこれらを試せば、その最小値が答えとなる

計算量は全体で $O(|X|^2 |Y|)$

ジャッジ解

- climpet (C++): 50行, 1077 bytes
- not (C++): 54行, 1284 bytes
- smiken (C++): 68行, 1037 bytes
- hos (Java): 65行, 1785 bytes

統計情報

- AC / trying teams
  - 43 / 80
- First acceptance
  - 非現役込み: Heno World (45:04)
  - 現役のみ: Heno World (45:04)