

## Mutujące DNA (Mutating DNA)

Grażyna jest biologką pracującą w firmie bioinformatycznej w Singapurze. Częścią jej pracy jest analiza sekwencji DNA różnych organizmów. Sekwencja DNA jest zdefiniowana jako napis złożony ze znaków "A", "T", oraz "C". Zwróć uwagę na to, że w tym zadaniu sekwencje DNA **nie zawierają "G"**.

Definiujemy mutację jako operację na sekwencji DNA polegającą na zamianie dwóch elementów sekwencji. Na przykład, pojedyncza mutacja może przekształcić "ACTA" w "AATC" poprzez zamianę dwóch zaznaczonych znaków "A" oraz "C".

Odległość mutacyjna między dwoma sekwencjami jest najmniejszą liczbą mutacji potrzebnych do przekształcenia jednej sekwencji w drugą, lub  $-1$  gdy nie jest możliwe przekształcenie jednej sekwencji w drugą za pomocą ciągu mutacji.

Grażyna analizuje dwie sekwencje DNA  $a$  oraz  $b$ , obie złożone z  $n$  elementów indeksowanych od  $0$  do  $n - 1$ . Twoim zadaniem jest pomóc Grażynie w znalezieniu odpowiedzi na  $q$  pytań postaci: jaka jest odległość mutacyjna między fragmentem  $a[x..y]$  oraz fragmentem  $b[x..y]$ ? Przez fragment  $s[x..y]$  sekwencji DNA  $s$  rozumiemy tutaj ciąg kolejnych znaków  $s$  o indeksach od  $x$  do  $y$  (włącznie). Inaczej mówiąc,  $s[x..y]$  jest sekwencją  $s[x]s[x + 1] \dots s[y]$ .

### Szczegóły implementacyjne

Powinieneś zaimplementować następujące procedury:

```
void init(string a, string b)
```

- $a$ ,  $b$ : napisy długości  $n$  opisujące dwie sekwencje DNA, które będą analizowane.
- Ta procedura będzie wywołana dokładnie raz, przed wszystkimi wywołaniami `get_distance`.

```
int get_distance(int x, int y)
```

- $x$ ,  $y$ : początkowy i końcowy indeks fragmentów, która należy przeanalizować.
- Wynikiem działania procedury powinna być odległość mutacyjna między fragmentami  $a[x..y]$  oraz  $b[x..y]$ .
- Procedura będzie wywołana dokładnie  $q$  razy.

### Przykład

Rozważ następujące wywołanie:

```
init("ATACAT", "ACTATA")
```

Powiedzmy, że sprawdzaczka wywoła `get_distance(1, 3)`. Wynikiem takiego wywołania powinna być odległość mutacyjna między  $a[1..3]$  oraz  $b[1..3]$ , czyli sekwencjami "TAC" oraz "CTA". Sekwencja "TAC" może być przekształcona w "CTA" przy użyciu 2 mutacji: **TAC** → **CAT**, a następnie **CAT** → **CTA**, a jednocześnie nie może być przekształcona przy użyciu mniej niż 2 mutacji.

Wynikiem wywołania powinno więc być 2.

Powiedzmy, że sprawdzaczka wywoła `get_distance(4, 5)`. Wynikiem takiego wywołania powinna być odległość mutacyjna między sekwencjami "AT" oraz "TA". Sekwencja "AT" może być przekształcona w "TA" przy użyciu pojedynczej mutacji, a jednocześnie jest jasne, że potrzebna jest przynajmniej jedna mutacja.

Wynikiem wywołania powinno więc być 1.

Na koniec powiedzmy, że sprawdzaczka wywoła `get_distance(3, 5)`. Ponieważ **nie jest możliwe** przekształcenie sekwencji "CAT" w "ATA" za pomocą ciągu mutacji, wynikiem takiego wywołania powinno być -1.

## Ograniczenia

- $1 \leq n, q \leq 100\,000$
- $0 \leq x \leq y \leq n - 1$
- Każdy znak  $a$  oraz  $b$  to "A", "T", lub "C".

## Podzadania

1. (21 punktów)  $y - x \leq 2$
2. (22 punkty)  $q \leq 500$ ,  $y - x \leq 1000$ , każdy znak  $a$  oraz  $b$  to "A" lub "T".
3. (13 punktów) każdy znak  $a$  oraz  $b$  to "A" lub "T".
4. (28 punktów)  $q \leq 500$ ,  $y - x \leq 1000$
5. (16 punktów) Brak dodatkowych ograniczeń.

## Przykładowa sprawdzaczka

Przykładowa sprawdzaczka wczytuje dane w następującym formacie:

- wiersz 1:  $n\ q$
- wiersz 2:  $a$
- wiersz 3:  $b$
- wiersz  $4 + i$  ( $0 \leq i \leq q - 1$ ):  $x\ y$  dla  $i$ -tego wywołania `get_distance`.

Przykładowa sprawdzaczka wypisuje Twoje odpowiedzi w następującym formacie:

- wiersz  $1 + i$  ( $0 \leq i \leq q - 1$ ): wynik  $i$ -tego wywołania `get_distance`.