

# ADN mutante

Grace es una bióloga que trabaja en una empresa de bioinformática en Singapur. Como parte de su trabajo, analiza las secuencias de ADN de varios organismos. Una secuencia de ADN se define como una cadena formada por los caracteres "A", "T" y "C". Tenga en cuenta que en esta tarea las secuencias de ADN **no contienen el carácter "G"**.

Nosotros definimos una mutación como una operación sobre una secuencia de ADN en la que se intercambian dos elementos de la secuencia. Por ejemplo, una sola mutación puede transformar "ACTA" en "AATC" intercambiando los caracteres resaltados "A" y "C".

La distancia de mutación entre dos secuencias es el número mínimo de mutaciones necesarias para transformar una secuencia en la otra, o  $-1$  si no es posible transformar una secuencia en la otra usando mutaciones.

Grace está analizando dos secuencias de ADN  $a$  y  $b$ , ambas formadas por  $n$  elementos con índices de  $0$  a  $n - 1$ .

Tu tarea es ayudar a Grace a responder  $q$  preguntas de la forma: ¿cuál es la distancia de mutación entre la subcadena  $a[x..y]$  y la subcadena  $b[x..y]$ ? Aquí, una subcadena  $s[x..y]$  de una secuencia de ADN  $s$  se define como una secuencia de caracteres consecutivos de  $s$ , cuyos índices son  $x$  e  $y$  inclusive. En otras palabras,  $s[x..y]$  es la secuencia  $s[x]s[x + 1] \dots s[y]$ .

## Detalles de la implementación

Usted debe implementar los siguientes procedimientos:

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

- $a$ ,  $b$ : cadenas de longitud  $n$ , que describen las dos secuencias de ADN que se van a analizar.
- Este procedimiento se llama exactamente una vez, antes de cualquier llamada a `get_distance`.

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

- $x$ ,  $y$ : índices inicial y final de las subcadenas a analizar.
- El procedimiento debe devolver la distancia de mutación entre las subcadenas  $a[x..y]$  y  $b[x..y]$ .
- Este procedimiento se llama exactamente  $q$  veces.

## Ejemplo

Considere la siguiente llamada:

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

Digamos que el calificador llama a `get_distance(1, 3)`. Esta llamada debería devolver la distancia de mutación entre  $a[1..3]$  y  $b[1..3]$ , es decir, las secuencias "TAC" y "CTA". "TAC" puede transformarse en "CTA" mediante 2:mutaciones **TAC**  $\rightarrow$  **CAT**, seguido de **CAT**  $\rightarrow$  **CTA**, y la transformación es imposible con menos de 2 mutaciones.

Por lo tanto, esta llamada debería devolver 2.

Digamos que el calificador llama a `get_distance(4, 5)`. Esta llamada debería devolver la distancia de mutación entre las secuencias "AT" y "TA". "AT" puede transformarse en "TA" a través de una sola mutación, y claramente se requiere al menos una mutación.

Por lo tanto, esta llamada debería devolver 1.

Finalmente, digamos que el calificador llama a `get_distance(3, 5)`. Dado que no hay **ninguna manera** de que la secuencia "CAT" se transforme en "ATA" a través de cualquier secuencia de mutaciones, esta llamada debería devolver  $-1$ .

## Restricciones

- $1 \leq n, q \leq 100\,000$
- $0 \leq x \leq y \leq n - 1$
- Cada carácter de  $a$  y  $b$  es uno de los caracteres "A", "T" y "C"

## Subtareas

1. (21 puntos)  $y - x \leq 2$
2. (22 puntos)  $q \leq 500$ ,  $y - x \leq 1000$ , cada carácter de  $a$  y  $b$  es o bien "A" o "T".
3. (13 puntos) cada carácter de  $a$  y  $b$  es o bien "A" o "T".
4. (28 puntos)  $q \leq 500$ ,  $y - x \leq 1000$
5. (16 puntos) No hay restricciones adicionales.

## Ejemplo del calificador

El calificador de ejemplo lee la entrada en el siguiente formato:

- línea 1:  $n \ q$
- línea 2:  $a$
- línea 3:  $b$
- línea  $4 + i$  ( $0 \leq i \leq q - 1$ ):  $x \ y$  para el  $i$ -th llamada por `get_distance`.

El calificador de ejemplo imprime tus respuestas en el siguiente formato:

- línea  $1 + i$  ( $0 \leq i \leq q - 1$ ) : el valor de retorno del  $i$ -th llamado por `get_distance`.