

Мутація ДНК

Біолог Галя працює у біоінформатичній компанії у Сінгапурі. Один з її обов'язків - аналіз ДНК послідовностей різних організмів. Послідовність ДНК визначається як рядок, що містить символи "A", "T", та "C". Зауважте, що у цій задачі послідовності ДНК **не містять символ "G"**.

Мутацією є операція над послідовністю ДНК, коли два елементи обмінюються своїми місцями. Наприклад, одна мутація може перетворити "ACTA" у "AATC" міняючи виділені символи "A" та "C" місцями.

Мутаційною відстанню між двома послідовностями є мінімальна кількість мутацій що потрібна для передворення однієї послідовності у другу, або -1 , якщо немає можливості перетворити одну послідовність у іншу за допомогою мутацій.

Галя аналізує дві послідовності ДНК a та b , кожна з яких складається з n елементів з індексами від 0 до $n - 1$. Ваша задача - допомогти Галі відповісти на q запитань вигляду: яка мутаційна відстань між підрядком $a[x..y]$ та підрядком $b[x..y]$? Підрядок $s[x..y]$ послідовності ДНК s визначається як послідовність сусідніх символів s , з індексами від x до y включно. Іншими словами, $s[x..y]$ є послідовністю $s[x]s[x + 1] \dots s[y]$.

Деталі реалізації

Ви маєте реалізувати наступну процедуру:

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

- a , b : рядки довжини n , що задають дві послідовності ДНК для аналізу.
- Ця процедура викликається рівно один раз, перед усіма викликами `get_distance`.

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

- x , y : початковий і кінцевий індекси підрядків для аналізу.
- Процедура має повертати мутаційну відстань між підрядками $a[x..y]$ та $b[x..y]$.
- Процедура викликається рівно q разів.

Приклад

Розглянемо наступний виклик:

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

Нехай модуль перевірки викликає `get_distance(1, 3)`. Цей виклик має повернути мутаційну відстань між $a[1..3]$ та $b[1..3]$, тобто між послідовностями "ТАС" та "СТА". "ТАС" можна перетворити на "СТА" за допомогою 2 мутацій: **ТАС** \rightarrow **САТ**, після чого **САТ** \rightarrow **СТА**, та перетворення неможливе якщо мутацій менше ніж 2.

Отже, виклик має повернути 2.

Нехай тепер модуль перевірки викликає `get_distance(4, 5)`. Цей виклик має повернути мутаційну відстань між послідовностями "АТ" та "ТА". "АТ" можна перетворити на "ТА" однією мутацією, та очевидно що потрібна принаймні одна мутація.

Отже, виклик має повернути 1.

Зрештою, нехай модуль перевірки викликає `get_distance(3, 5)`. Оскільки **немає способу** перетворити "САТ" на "АТА" використовуючі мутації, цей виклик має повернути -1 .

Обмеження

- $1 \leq n, q \leq 100\,000$
- $0 \leq x \leq y \leq n - 1$
- Кожен символ у послідовностях a та b - один з "А", "Т", та "С".

Підзадачі

1. (21 бал) $y - x \leq 2$
2. (22 бали) $q \leq 500$, $y - x \leq 1000$, кожен символ a та b є або "А", або "Т".
3. (13 балів) кожен символ a та b є або "А", або "Т".
4. (28 балів) $q \leq 500$, $y - x \leq 1000$
5. (16 балів) без додаткових обмежень.

Приклад модуля перевірки

Приклад модуля перевірки читає вхідні дані у наступному форматі:

- рядок 1: $n\ q$
- рядок 2: a
- рядок 3: b
- рядок $4 + i$ ($0 \leq i \leq q - 1$): $x\ y$ для i -го виклику `get_distance`.

Приклад модуля перевірки друкує ваші відповіді у наступному форматі:

- рядок $1 + i$ ($0 \leq i \leq q - 1$): значення, що повертає i -й виклик `get_distance`.