

Mutating DNA

Биолог изучава ДНК на различни организми. ДНК е последователност от знаците "A", "T" и "C". Специално за тази задача ДНК **не съдържа знака "G"**.

Дефинираме мутация като операция върху ДНК последователност, при която два елемента си разменят местата. Например една мутация трансформира "АСТА" в "ААТС" и това става чрез разменяне на местата на знаците, отбелязани с получен шрифт "A" и "C".

Мутационно разстояние между две редици наричаме най-малкия брой мутации, чрез които едната редица се трансформира в другата, а ако не е възможно да се направи такава трансформация чрез мутации, разстоянието го дефинираме да е -1 .

Биологът анализира две ДНК последователности a и b , като и двете съдържат n елемента, които индексирате с индекси от 0 до $n - 1$. Вашата задача е да помогнете, като отговорите на q въпроса от вида: колко е мутационното разстояние между поднизовите $a[x..y]$ и $b[x..y]$? Тук поднизът $s[x..y]$ от ДНК последователността s се дефинира като редица от последователните знаци от s , чиито индекси са от x до y включително. т.е. $s[x..y]$ е редицата $s[x]s[x + 1] \dots s[y]$.

Имплементация

Трябва да имплементирате следните функции:

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

- a , b : низове с дължина n , описващи две ДНК последователности.
- Тази функция се извиква точно веднъж, преди извикванията на `get_distance`.

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

- x , y : началният и крайният индекс на поднизовите, които се анализират.
- Тази функция трябва да върне мутационното разстояние между поднизовите $a[x..y]$ и $b[x..y]$.
- Тази функция се извиква точно q пъти.

Пример

Разглеждаме следното извикване:

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

Нека грейдерът извика `get_distance(1, 3)`.

Това извикване трябва да върне мутационното разстояние между $a[1..3]$ и $b[1..3]$, т.е. между низовете "TAC" и "CTA".

Низът "TAC" може да бъде трансформиран в "CTA" чрез 2 мутации: Мутацията **TAC** \rightarrow **CAT**, следвана от **CAT** \rightarrow **CTA**, и не е възможно това да стане с по-малко от 2 мутации.

Следователно, това извикване трябва да върне 2.

Нека грейдерът извика `get_distance(4, 5)`.

Това извикване трябва да върне мутационното разстояние между "AT" и "TA". Низът "AT" може да бъде трансформиран в "TA" чрез една мутация, очевидно.

Следователно това извикване трябва да върне 1.

Нека грейдерът извика `get_distance(3, 5)`.

Понеже **не е възможно** низът "CAT" да бъде трансформиран в "ATA", това извикване трябва да върне -1 .

Ограничения

- $1 \leq n, q \leq 100\,000$
- $0 \leq x \leq y \leq n - 1$
- Всеки знак на a и b е един от знаците "A", "T", или "C".

Подзадачи

1. (21 точки) $y - x \leq 2$
2. (22 точки) $q \leq 500$, $y - x \leq 1000$, всеки знак на a и b е или "A", или "T".
3. (13 точки) a и b съдържат само знаците "A" и "T".
4. (28 точки) $q \leq 500$, $y - x \leq 1000$
5. (16 точки) Няма допълнителни ограничения.

Примерен грейдер

Примерният грейдер чете вход в следния формат:

- ред 1: $n\ q$
- ред 2: a
- ред 3: b
- ред $4 + i$ ($0 \leq i \leq q - 1$): $x\ y$ за i -тото извикване на `get_distance`.

Примерният греjder отпечатва отговора в следния формат:

- ред $1 + i$ ($0 \leq i \leq q - 1$): стойността, която връща i -тото извикване на `get_distance`.