

# Mutating DNA

Grace je biolog koji radi u firmi za bioinformatiku u Singapuru. Kao dio svog posla, ona analizira DNK sekvence različitih organizama. DNK sekvenca je definisana kao string koji se sastoji od karaktera "A", "T", and "C". Imajte na umu da ovom zadatku DNK sekvenca **ne sadrže karakter "G"**.

Definisaćemo mutaciju kao jednu operaciju na jednoj DNA sekvenci tokom koje dva elementa te sekvence zamijene mjesta. Na primjer, jedna mutacija može transformisati "ACTA" u "AATC" izmjenjujući mjesta isključivih karaktera "A" i "C".

Udaljenost mutiranja između dvije sekvence je minimalan broj mutacija koje treba izvršiti da transformišemo jednu sekvencu u drugu, ili  $-1$  ukoliko nije moguće preći iz jedne u drugu sekvencu koristeći isključivo mutacije.

Grace analizira dvije DNA sekvence,  $a$  i  $b$ , i obje se sastoje od ukupno  $n$  elemenata indeksiranih od 0 do  $n - 1$ . Vaš zadatak je da pomognete Grace da odgovori na  $q$  pitanja: koja je udaljenost mutiranja između datog podstringa  $a[x..y]$  i datog podstringa  $b[x..y]$ ? Ovdje imamo da je podstring  $s[x..y]$  jedne DNA sekvence  $s$  definisan kao sekvenca uzastopnih karaktera stringa  $s$ , sa indeksima  $x$  i  $y$ , uključivo. Drugim riječima,  $s[x..y]$  je sekvenca  $s[x]s[x + 1] \dots s[y]$ .

## Detalji implementacije

Potrebno je implementirati sljedeću proceduru:

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

- $a$ ,  $b$ : stringovi dužine  $n$  koji opisuju dvije DNA sekvence koje treba analizirati.
- Ova se procedura poziva tačno jednom, prije bilo kojeg poziva procedure `get_distance`.

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

- $x$ ,  $y$ : početni i kranji indeksi substringova koje treba analizirati.
- Ova procedura treba da vrati udaljenost mutiranja između dva podstringa  $a[x..y]$  i  $b[x..y]$ .
- Ova procedura se poziva tačno  $q$  puta.

## Primjer

Razmotrimo sljedeći poziv:

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

Recimo sada da grader poziva `get_distance(1, 3)`. Ovaj poziv treba da vrati udaljenost mutiranja između  $a[1..3]$  i  $b[1..3]$ , to jest, između sekvenci "TAC" i "CTA". "TAC" se može transformirati u "CTA" pomoću 2 mutacije: **TAC** → **CAT**, pa onda **CAT** → **CTA**, a nije moguće imati transformaciju sa manje od 2 mutacije.

Prema tome, ovaj poziv treba da vrati 2.

Recimo sada da grader poziva `get_distance(4, 5)`. Ovaj poziv treba da vrati udaljenost mutiranja između sekvenci "AT" i "TA". "AT" se može transformirati u "TA" pomoću jedne mutacije, i jasno je da je potrebna najmanje jedna mutacija.

Prema tome, ovaj poziv treba da vrati 1.

Na kraju, recimo da grader poziva `get_distance(3, 5)`. Kako **ne postoji način** da sekvencu "CAT" transformišemo u "ATA" pomoću bilo kojeg niza mutacija, ovaj poziv treba da vrati  $-1$ .

## Ograničenja

- $1 \leq n, q \leq 100\,000$
- $0 \leq x \leq y \leq n - 1$
- Svaki karakter stringova  $a$  i  $b$  je ili "A", ili "T", ili "C".

## Podzadaci

1. (21 bod)  $y - x \leq 2$
2. (22 boda)  $q \leq 500$ ,  $y - x \leq 1000$ ,  $a$  i  $b$  se sastoje samo od karaktera "A" i "T".
3. (13 bodova)  $a$  i  $b$  se sastoje samo od karaktera "A" i "T".
4. (28 bodova)  $q \leq 500$ ,  $y - x \leq 1000$
5. (16 bodova) Nema dodatnih ograničenja.

## Sample grader

Sample grader čita ulaz u sljedećem formatu:

- linija 1:  $n\ q$
- linija 2:  $a$
- linija 3:  $b$
- linija  $4 + i$  ( $0 \leq i \leq q - 1$ ):  $x\ y$  za  $i$ -ti poziv procedure `get_distance`.

Sample grader štampa odgovore u sljedećem formatu:

- linija  $1 + i$  ( $0 \leq i \leq q - 1$ ): vraćena vrijednost  $i$ -tog poziva procedure `get_distance`.