

# მეტაციური დნმ

გრეისი ბიოლოგია და სინგაპურის ბიოინფორმატიკის კომპანიაში მუშაობს. სამსახურში მას სხვადასხვა ორგანიზმების დნმ მიმდევრობების ანალიზი ევალება. **დნმ** მიმდევრობა განისაზღვრება როგორც "A", "T" და "C" სიმბოლოებისაგან შედგენილი სტრიქონი. ყურადღება მიაქციეთ იმას, რომ ამ ამოცანაში დნმ მიმდევრობები არ შეიცავენ "G" სიმბოლოს.

განვსაზღვროთ მეტაცია როგორც დნმ მიმდევრობაზე ჩატარებული ისეთი ოპერაცია, რომლის დროსაც ამ მიმდევრობის ორი ელემენტი ცვლის ადგილებს. მაგალითად, ერთი მეტაციით სტრიქონი "ACTA" გარდაიქმნება სტრიქონად "AATC" მეტი შრიფტით გამოყოფილი "A" და "C" სიმბოლოების ადგილების გაცვლით.

ორ მიმდევრობას შორის მეტაციური მანძილი ეწოდება მეტაციების იმ მინიმალურ რაოდენობას, რომელიც საჭიროა ერთი მათგანის მეორედ გარდასაქმნელად. თუ ასეთი გარდაქმნა შეუძლებელია, მაშინ მეტაციური მანძილი -1-ის ტოლია.

გრეისი ანალიზს უკეთებს ორ  $a$  და  $b$  დნმ მიმდევრობას. ორივე მათგანი შედგება 0-დან  $(n - 1)$ -მდე გადანომრილი  $n$  რაოდენობის ელემენტისაგან. თქვენი ამოცანაა დაეხმაროთ გრეისს უპასუხოს  $q$  რაოდენობის შემდეგი სახის შეკითხვას: რისი ტოლია მეტაციური მანძილი  $a[x..y]$  ქვესტრიქონსა და  $b[x..y]$  ქვესტრიქონს შორის? აქ  $s$  დნმ მიმდევრობის  $s[x..y]$  ქვესტრიქონი განისაზღვრება, როგორც  $s$ -ის მიმდევრობით აღებული,  $x$ -დან  $y$ -მდე (ჩათვლით) ინდექსების მქონე სიმბოლოთა მიმდევრობა. სხვა სიტყვებით,  $s[x..y]$  წარმოადგენს მიმდევრობას:  $s[x]s[x + 1] \dots s[y]$ .

## იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია:

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

- $a$ ,  $b$ : სტრიქონები სიგრძით  $n$ , აღწერენ იმ ორ დნმ მიმდევრობას, რომელთა ანალიზიც უნდა მოხდეს;
- ამ პროცედურის გამოძახება ხდება მხოლოდ ერთხელ, `get_distance`-ის ნებისმიერი გამოძახების წინ.

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

- $x$ ,  $y$ : იმ ქვესტრიქონების საწყისი და საბოლოო ინდექსები, რომელთა ანალიზიც უნდა მოხდეს;

- პროცედურამ უნდა დააბრუნოს  $T$   $a[x..y]$  და  $b[x..y]$  ქვესტრიქონებს შორის მუტაციური მანძილი;
- ამ პროცედურის გამოძახება ხდება ზუსტად  $q$ -ჯერ.

## მაგალითი

განვიხილოთ შემდეგი გამოძახება:

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

ვთქვათ გრადერი იძახებს `get_distance(1, 3)`. ამ გამოძახებამ უნდა დააბრუნოს მუტაციური მანძილი  $a[1..3]$ -სა და  $b[1..3]$ -ს შორის, ანუ "TAC"-სა და "CTA"-ს შორის. "TAC"-ის "CTA"-ად გარდასაქმნელად საჭიროა 2 მუტაცია: **TAC** → **CAT** და **CAT** → **CTA**. ამასთან, 2-ზე ნაკლები რაოდენობის მუტაციით ამ გარდაქმნის გაკეთება შეუძლებელია.

შესაბამისად, ამ გამოძახებამ უნდა დააბრუნოს 2.

ვთქვათ გრადერი იძახებს `get_distance(4, 5)`. ამ გამოძახებამ უნდა დააბრუნოს მუტაციური მანძილი "AT"-სა და "TA"-ს შორის. "AT"-ს "TA"-ად გარდაქმნა შესაძლებელია 1 მუტაციით და, ცხადია, რომ ეს ერთი მაინც მუტაცია საჭიროა.

შესაბამისად, ამ გამოძახებამ უნდა დააბრუნოს 1.

საბოლოოდ, ვთქვათ გრადერი იძახებს `get_distance(3, 5)`. რადგანაც "CAT"-ის "ATA"-ად გარდაქმნა შეუძლებელია, ამიტომ ამ გამოძახებამ უნდა დააბრუნოს  $-1$ .

## შეზღუდვები

- $1 \leq n, q \leq 100\,000$
- $0 \leq x \leq y \leq n - 1$
- $a$ -ს და  $b$ -ს თითოეული სიმბოლო წარმოადგენს ერთ-ერთს "A", "T" და "C" სიმბოლოებიდან.

## ქვეამოცანები

1. (21 ქულა)  $y - x \leq 2$
2. (22 ქულა)  $q \leq 500$ ,  $y - x \leq 1000$ ,  $a$  და  $b$  მხოლოდ "A" ან "T" სიმბოლოებისაგან შედგება;
3. (13 ქულა)  $a$  და  $b$  მხოლოდ "A" ან "T" სიმბოლოებისაგან შედგება;
4. (28 ქულა)  $q \leq 500$ ,  $y - x \leq 1000$
5. (16 ქულა) დამატებითი შეზღუდვების გარეშე.

## სანიმუშო გრადერი

სანიმუშო გრადერს შეაქვს მონაცემები შემდეგი ფორმატით:

- სტრიქონი 1:  $n$   $q$

- სტრიქონი 2:  $a$
- სტრიქონი 3:  $b$
- სტრიქონი  $4 + i$  ( $0 \leq i \leq q - 1$ ):  $x$   $y$  პროცედურა `get_distance`-ის  $i$ -ური გამოძახებისათვის.

სანიშნო გრაფერს გამოაქვს თქვენი პასუხები შემდეგი ფორმატით:

- სტრიქონი  $1 + i$  ( $0 \leq i \leq q - 1$ ): დასაბრუნებელი მნიშვნელობა `get_distance`-ის  $i$ -ური გამოძახებისათვის.