

DNA mutácie (Mutating DNA)

Gréta robí v labáku v Šarišských Michaľanoch. V rámci svojej práce analyzuje DNA sekvencie rôznych organizmov. Pre skúmané organizmy platí, že ich DNA sa dá popísať reťazcom obsahujúcim len znaky "A", "T" a "C". Všimnite si, že v tejto úlohe DNA sekvencie **nikdy neobsahujú znak "G"**.

Mutácia je operácia na DNA sekvencii, pri ktorej vymeníme niektoré dva jej prvky. Napríklad jedna mutácia môže z "ACTA" vyrobiť "AATC" tak, že si zvýraznené znaky "A" a "C" vymenia pozíciu.

Mutačná vzdialenosť dvoch sekvencií je rovná minimálnemu počtu postupných mutácií, ktorý je potrebný na transformáciu jednej postupnosti na druhú. Ak toto vôbec nie je možné dosiahnuť, mutačná vzdialenosť je rovná -1 .

Gréta práve analyzuje dve DNA sekvencie a a b . Obe majú rovnakú dĺžku: n znakov. V každej sekvencii majú písmená, ktoré ju tvoria, indexy od 0 po $n - 1$.

Tvojou úlohou je zistiť Gréte odpovede na q otázok. Každá otázka je tvaru: "Akú mutačnú vzdialenosť majú podreťazce $a[x..y]$ a $b[x..y]$?"

Značenie $s[x..y]$ predstavuje úsek DNA sekvencie s tvorený znakmi na indexoch x až y , vrátane oboch koncov. Inými slovami, ide o postupnosť znakov $s[x]s[x + 1] \dots s[y]$.

Detaily implementácie

Tvojou úlohou je implementovať nasledujúce funkcie:

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

- a , b : reťazce dĺžky n predstavujúce analyzované DNA sekvencie.
- Túto funkciu grader zavolá práve raz, a to skôr ako spraví ľubovoľné volanie funkcie `get_distance`.

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

- x , y : index začiatku a konca skúmaného úseku v oboch reťazcoch.
- Táto funkcia by mala vrátiť mutačnú vzdialenosť reťazcov $a[x..y]$ a $b[x..y]$.
- Túto funkciu grader postupne zavolá q -krát.

Príklad

Uvažujme nasledovné volanie funkcie `init`:

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

Povedzme, že grader následne zavolá `get_distance(1, 3)`. Toto volanie by malo vrátiť mutačnú vzdialenosť medzi $a[1..3]$ a $b[1..3]$. Toto sú reťazce "TAC" a "CTA". Ich mutačná vzdialenosť je 2, lebo na 2 mutácie vieme prvý reťazec zmeniť na druhý (napr. **TAC** → **CAT** a následne **CAT** → **CTA**) a na menej ako 2 mutácie to nejde. Toto volanie by teda malo vrátiť hodnotu 2.

Ďalej sa pozrime na volanie `get_distance(4, 5)`. Toto volanie sa pýta na mutačnú vzdialenosť reťazcov "AT" a "TA", čo je zjavne 1. Toto volanie by teda malo vrátiť hodnotu 1.

Na záver sa pozrime na volanie `get_distance(3, 5)`. Keďže mutáciami **sa nedá** prerobiť reťazec "CAT" na reťazec "ATA", toto volanie má vrátiť -1 .

Obmedzenia

- $1 \leq n, q \leq 100\,000$
- $0 \leq x \leq y \leq n - 1$
- Každý znak v a aj b je "A", "T", alebo "C".

Subtasks

1. (21 points) $y - x \leq 2$
2. (22 points) $q \leq 500$, $y - x \leq 1000$ a každý znak v a aj b je "A" alebo "T".
3. (13 points) každý znak v a aj b je "A" alebo "T".
4. (28 points) $q \leq 500$, $y - x \leq 1000$
5. (16 points) bez ďalších obmedzení

Ukážkový grader

Ukážkový grader očakáva vstup v nasledovnom formáte:

- riadok 1: $n\ q$
- riadok 2: a
- riadok 3: b
- riadok $4 + i$ ($0 \leq i \leq q - 1$): $x\ y$ pre volanie `get_distance`, ktoré dostane poradové číslo i .

Ukážkový grader vypíše návratové hodnoty volaní tvojej funkcie `get_distance` v nasledovnom formáte:

- line $1 + i$ ($0 \leq i \leq q - 1$): návratová hodnota jej volania číslo i .