

Zadanie: DIN

Dinozaur

Laboratorium z ASD, Zadanie Zaliczeniowe 2. Dostępna pamięć: 256 MB. 30.12.2021, 23:59:59

Zarząd Parku Jurajskiego ma zmartwienie. Wpływy z biletów zaczynają maleć. Badania działu marketingu wykazały, że jest tak dlatego, iż zdaniem klientów dinozaury w parku nie wyglądają dość groźnie. W związku z tym zarząd parku zwrócił się do znanego paleontologa, profesora Makarego, o stworzenie kodu DNA nowego, groźniejszego dinozaura.

Profesor Makary słynie w środowisku naukowym z odkrytej przez siebie metody inżynierii genetycznej zwaną metodą prób i błędów. Postanowił zastosować ją i tym razem. Profesor zaczyna od kodu DNA najgroźniejszego dinozaura w parku, a następnie wielokrotnie modyfikuje kod za pomocą jednej z dwóch operacji: odwrócenia fragmentu kodu oraz przeniesienia fragmentu kodu w inne miejsce. Od czasu do czasu, profesor bada jak groźnie wyglądałby dinozaur odpowiadający aktualnemu kodowi genetycznemu. Odkrył on, że za groźny wygląd odpowiadają spójne ciągi tych samych liter w kodzie. Dlatego profesor chciałby znać najdłuższe takie ciągi w wybranych fragmentach kodu. Pomóż profesorowi i napisz program, który pozwoli mu stworzyć potwora!

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite: n ($1 \leq n \leq 1\,000\,000$) oraz m ($1 \leq m \leq 100\,000$).

Kolejny wiersz zawiera słowo s_0 długości n złożone z liter A, G, C i T.

Każdy z kolejnych m wierszy zawiera jedno z trzech poleceń. Po wykonaniu i -tego polecenia otrzymujemy słowo s_i , zgodnie z następującymi zasadami.

- $O\ j\ k$, ($1 \leq j \leq k \leq n$) oznacza, że słowo s_i powstaje z s_{i-1} poprzez odwrócenie fragmentu zaczynającego się od indeksu j , a kończącego na indeksie k . Przykładowo, jeśli $s_{i-1} = \text{AGCG}$, to wynikiem operacji $O\ 2\ 3$ będzie $s_i = \text{ACGG}$.
- $P\ j\ k\ l$, ($1 \leq j \leq k \leq n$, $1 \leq l \leq n - (k - j)$) oznacza, że słowo s_i powstaje z s_{i-1} poprzez usunięcie fragmentu zaczynającego się od indeksu j i kończącego na indeksie k otrzymując słowo w , a następnie wstawienie tego fragmentu do słowa w , między znaki o indeksach $l - 1$ oraz l . Przykładowo:
 - jeśli $s_{i-1} = \text{AGCT}$, to wynikiem operacji $P\ 1\ 2\ 2$ będzie $s_i = \text{CAGT}$,
 - jeśli $s_{i-1} = \text{AGCT}$, to wynikiem operacji $P\ 1\ 2\ 3$ będzie $s_i = \text{CTAG}$,
 - jeśli $s_{i-1} = \text{AGCT}$, to wynikiem operacji $P\ 2\ 3\ 1$ będzie $s_i = \text{GCAT}$.
- $N\ j\ k$, ($1 \leq j \leq k \leq n$) nie modyfikuje słowa, tzn. $s_i = s_{i-1}$. Natomiast po każdym takim poleceniu Twój program powinien wypisać na standardowe wyjście maksymalną liczbę kolejnych identycznych liter we fragmencie słowa s_{i-1} zaczynającym się od indeksu j a kończącym na indeksie k . Przykładowo, jeśli $s_{i-1} = \text{ATTT}$, to w wyniku operacji $N\ 1\ 3$ na wyjście wypisany zostanie liczba 2.

Wyjście

Dla każdej operacji postaci $N\ j\ k$ na wejściu, wyjście powinno zawierać jeden wiersz zawierający jedną liczbę całkowitą (maksymalną liczbę kolejnych identycznych liter we fragmencie słowa utworzonego przez wcześniejsze operacje, zaczynającym się od indeksu j a kończącym na indeksie k).

Przykład

Dla danych wejściowych:

```
5 6
AGCTA
O 2 5
N 1 3
P 2 3 3
N 1 5
P 1 2 2
N 1 5
```

Poprawnym wynikiem jest:

```
2
1
2
bo operacje modyfikacji tworzą kolejno słowa AATCG,
ACATG oraz AACTG.
```