



Изяждане на букви

Бидик харесва буквите. Той толкова много ги харесва, че чувства нужда от време на време да изяде по някоя буква. Бидик получава матрица (с големина $n \times m$) от главни латински букви. Всяка буква е или старо ('O'), или ново ('N'). Старите букви са от 2023 и носят 2023 калории, докато новите букви са от тази година (2024) и носят 2024 калории (всяка буква носи 2023 или 2024 калории). Бидик също има известна сила на скачане - цяло число k . Той не може да скача твърде далеч по буквите, така че ще изяде буквите според тези правила:

- Бидик има предпочитания за първата и последната буква. Затова той трябва винаги да започне пира си от буквата на позиция $(0, 0)$ и да го приключи на позиция $(n - 1, m - 1)$.
- След като изяде буква на позиция (i, j) , той може да изяде буква на всяка позиция (a, b) (различна от (i, j)), така че:
 - $i \leq a$
 - $j \leq b$
 - $|i - a| + |j - b| \leq k$

Тези правила означават, че Бидик не може да изяде буква, която е много "далеч" от предната.

- Той единствено може да изяде буквите поред, т.е. след A той може да изяде само B, след B той може да изяде само C и т.н. След Z той може да изяде само A.
- Изядането на стара буква носи 2023 калории, докато изядането на нова буква носи 2024 калории.

Така всяка възможна поредица от изядания на букви има цена, която се смята като сумата на броя калории на изядените букви. Ако няма валидни поредици от изядания, то максималната стойност, която Бидик може да получи е 0.

Миле не иска Бидик да има голямо шкембе, затова той се опитва да го спре да изяде букви. Той има q възможни действия. За действие p , Миле ще избере буквата (x_p, y_p) и ще я промени на t_p . За всяко действие, той иска да знае, след като изпълни действието си, дали максималната стойност ще се увеличи (Лошо!), ще остане същата (Арно!) или ще намалее (Добре!). Обърнете внимание, че действията са независими едни от други, т.е. те не променят получената матрица от Бидик.

Детайли по имплементацията

Имплементирайте функцията за изяждане на букви, която е предоставена във файла `letter-eating.cpp`. Трябва да включите хедърния файл `letter-eating.h`.

Имплементирайте функцията

```
vector< char > letter_eating(int n, int m, int k, int q,  
                           vector< vector< char > > c,  
                           vector< vector< char > > v, vector< int > x,  
                           vector< int > y, vector< char > t)
```

предоставена във файла `letter-eating.cpp` със следните параметри:

- По-рано споменатите цели числа n, m, k, q ($1 \leq n, m, k, q \leq 10^5, 1 \leq nm \leq 10^5$).
- Две матрици с n реда и m колони. c_{ij} и v_{ij} ($0 \leq i \leq n - 1, 0 \leq j \leq m - 1$) описват буквата и възрастта на буквата на тази позиция, съответно. c_{ij} е главна латинска буква, а v_{ij} е или 'O' или 'N', както беше описано по-горе.
- Три масива с големина q , x_p, y_p, t_p ($0 \leq p \leq q - 1$), където всяка тройка (x_p, y_p, t_p) описва заявка, както беше обяснено по-горе.

Върнете един масив със символи с големина q , като всеки символ е равен на 'L' за увеличаваща максимална стойност, 'A' за запазване на максималната стойност или 'D' за намаляваща максимална стойност.

Локален грейдър

Локалният грейдър има следния формат на входа:

- Ред 1: $n \ m \ k \ q$
- Ред $2 + i$ (for $0 \leq i \leq n - 1$): m символа, $c_{i1} \ c_{i2} \ \dots \ c_{im}$
- Ред $2 + n + i$ (for $0 \leq i \leq n - 1$): m символа, $v_{i1} \ v_{i2} \ \dots \ v_{im}$
- Ред $2 + 2n + p$ (for $0 \leq p \leq q$): $x_p \ y_p \ t_p$

Пример

Извикване	Върната стойност
<code>letter_eating(3, 3, 2, 2, {{ A, A, B }, { D, B, C }, { D, B, D }}, {{ N, O, N }, { O, O, N }, { N, N, O }}, { 0, 2 }, { 0, 2 }, { Z, J })</code>	<code>{ L, D }</code>
<code>letter_eating(6, 1, 3, 5, {{ C }, { D }, { E }, { D }, { E }, { F }}, {{ N }, { N }, { N }, { O }, { O }, { O }}, { 3, 1, 3, 3, 0 }, { 0, 0, 0, 0, 0 }, { Z, F, X, Y, D })</code>	<code>{ A, D, A, A, D }</code>

Подзадачи

- **(14 точки)** Отговорите ще бъдат само 'L' или 'A', $n = 1, q = 1$.
- **(16 точки)** Отговорите ще бъдат само 'L' или 'A', $n = 1, k \leq 100$.
- **(10 точки)** Отговорите ще бъдат само 'L' или 'A', $n = 1$.
- **(11 точки)** $k \leq 100$.
- **(19 точки)** $q = 1$.
- **(30 точки)** Няма допълнителни ограничения.