

Estudiando Prefijos

Contribución de Lautaro Lasorsa

Descripción del problema

Lautaro quiere probar tus habilidades programando, y para eso te plantea el siguiente desafío:

Inicialmente, se tienen N palabras vacías (sin caracteres). Cada palabra se identifica con un entero desde 1 hasta N , y a medida que se realizan operaciones, estas palabras se van modificando. Se realizan Q operaciones, que pueden ser de los siguientes tipos:

- **AGREGAR:** Dado un entero i y un carácter c (letra minúscula del alfabeto inglés), agregar el carácter c al final de la palabra i ($1 \leq i \leq N$).
- **BORRAR:** Dados dos enteros i y j ($1 \leq i \leq N$), borrar las últimas j letras de la palabra número i . Se garantiza que la palabra tiene al menos j letras (el resultado puede ser una palabra vacía)
- **CAMBIAR:** Dados dos enteros i y j ($1 \leq i, j \leq N$), intercambiar las palabras en dichas posiciones.
- **COPIAR:** Dados dos enteros i y j ($1 \leq i, j \leq N$), hacer que la palabra número j se vuelva igual a la palabra número i .
- **PREGUNTAR:** Dados dos enteros i y j ($1 \leq i \leq j \leq N$), calcular la longitud del prefijo común más largo entre todas las palabras cuyos índices estén entre i y j , inclusive. Por ejemplo, si la respuesta a esta consulta fuera 3, eso significaría que las primeras 3 letras de todas las palabras numeradas entre i y j inclusive, coinciden. Pero que esto ya no es verdad para las 4 primeras letras.

Detalles de implementación

Debes implementar las funciones

- `inicializar(N, Q)`, una función que será llamada una única vez al inicio del programa, siendo los enteros N y Q la cantidad de palabras y cantidad de operaciones respectivamente. Por cada operación se llamará a una de las siguientes funciones según corresponda.
- `agregar(i, c)`, recibe el entero i y el carácter c y debe implementar la operación **AGREGAR**.
- `borrar(i, j)`, recibe los enteros i y j y debe implementar la operación **BORRAR**.
- `cambiar(i, j)`, recibe los enteros i y j y debe implementar la operación **CAMBIAR**.
- `copiar(i, j)`, recibe los enteros i y j y debe implementar la operación **COPIAR**.
- `preguntar(i, j)`, recibe los enteros i y j y debe implementar la consulta **PREGUNTAR**. Retorna un entero con la respuesta correspondiente.

Evaluador local

El evaluador local lee de la entrada estándar una línea con los valores N y Q . Luego de realizar la llamada `inicializar(N, Q)`, lee las Q operaciones. Por cada una, lee primero un entero t que indica el tipo de operación, y en cada caso:

- Si $t = 0$, lee el entero i y el carácter c y llama a la función `agregar(i, c)`
- Si $t = 1$, lee los enteros i y j y llama a la función `borrar(i, j)`
- Si $t = 2$, lee los enteros i y j y llama a la función `cambiar(i, j)`.

- Si $t = 3$, lee los enteros i y j y llama a la función `copiar(i, j)`.
- Si $t = 4$, lee los enteros i y j y llama a la función `preguntar(i, j)`. Escribe a la salida estándar una línea con el valor retornado por la función.

Cotas

- $1 \leq N, Q \leq 200.000$

Ejemplos

Si el evaluador local recibe la siguiente entrada:

```
3 15
0 1 a
0 1 a
0 1 a
4 1 3
0 2 b
0 3 a
0 3 a
0 3 a
4 1 2
2 2 3
4 1 2
3 2 3
4 1 3
1 2 2
4 1 3
```

Una implementación correcta podrá devolver:

```
0
0
3
3
1
```

En cambio, si recibe:

```
3 19
0 1 a
0 1 b
0 1 c
0 1 d
0 1 e
4 1 2
0 2 a
4 1 2
0 2 b
4 1 2
0 2 c
4 1 2
0 2 d
4 1 2
0 2 e
4 1 2
4 1 3
3 1 3
4 1 3
```

Podrá devolver:

```
0
1
2
3
4
5
0
5
```

Subtareas

1. $1 \leq N, Q \leq 3.000$ y no hay llamadas a la función `copiar` (11 puntos)
2. $1 \leq N \leq 200.000$ y $1 \leq Q \leq 3.000$ (23 puntos)
3. Hay a lo sumo 100 llamadas a la función `preguntar`, y en todas las llamadas a la función `borrar` se tiene que $j = 1$ (18 puntos)
4. Hay a lo sumo 100 llamadas a la función `preguntar` (10 puntos)
5. Sin más restricción (38 puntos)