

## Construyendo Pueblos

Contribución de Lautaro Lasorsa

### Descripción del problema

El reino de Arbolonia está colonizando una nueva isla, y para eso busca construir pueblos en ella. El procedimiento para construir un pueblo es el siguiente:

1. Primero, se llega al único puerto que hay en la isla y se desembarca. En este momento se elige un número del 0 al 9, que indica en qué dirección se seguirá. El 0 es la dirección más a la izquierda y el 9 la dirección más a la derecha.
2. Después de recorrer una cierta distancia por el camino elegido, se vuelve a repetir la elección, nuevamente eligiendo una dirección entre 0 y 9 para seguir.
3. Después de  $M$  elecciones, se llega al punto en el que se construirá el nuevo pueblo.

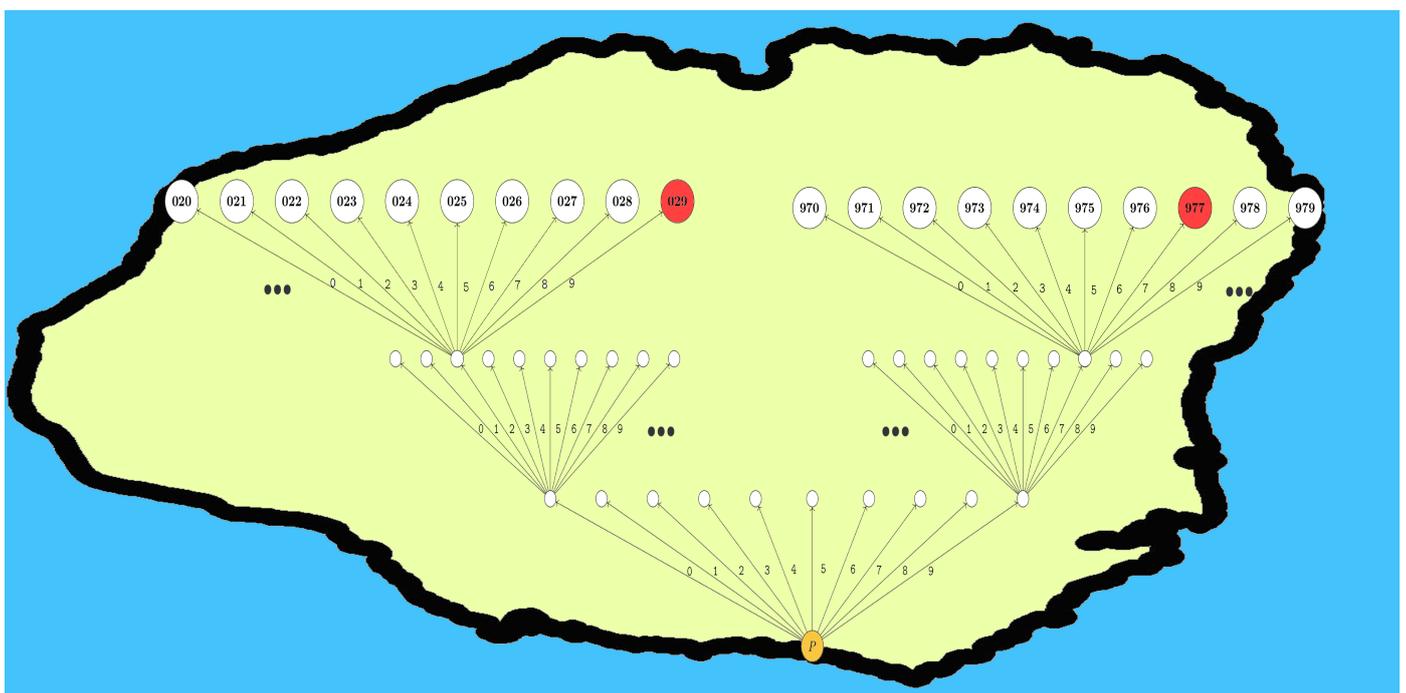
Por las características de este procedimiento, si para dos pueblos coinciden las primeras  $i$  elecciones, y en la elección  $i + 1$  para el primero se toma una ruta más a la derecha que la que se toma para el segundo, entonces el primero quedará más a la derecha sin importar las siguientes elecciones.

Dicho de otra forma: si  $A$  es una cadena de texto que indica las elecciones tomadas en la construcción del primer pueblo, y  $B$  la correspondiente cadena para el segundo pueblo; el pueblo más a la derecha será el que corresponda a la cadena lexicográficamente mayor. Estas cadenas son las que se utilizan para identificar unívocamente cada pueblo.

El rey de Arbolonia desea monitorear la colonización en tiempo real. La colonización involucra dos operaciones:

1. Crear un cierto pueblo identificado con la cadena  $A$ , con una población total  $P_A$ . Si el pueblo  $A$  ya existía, lo que hace esta operación es reportar el resultado de un censo, actualizando la población del pueblo al nuevo valor.
2. Consultar cuál es la población total de los pueblos existentes que quedan comprendidos entre  $A$  y  $B$ , inclusive; es decir, se debe contabilizar la población total que vive a la derecha de  $A$  y a la izquierda de  $B$  (visto desde el puerto). Las ubicaciones  $A$  y  $B$  pueden o no contener un pueblo.

Tu trabajo es implementar el sistema informático necesario para responder las preguntas de su majestad.



### Detalles de implementación

Debes implementar las funciones

- `inicializar(Q, M)`, una función que será llamada una única vez al inicio del programa. Sus parámetros son:
  - $Q$ , la cantidad de llamadas que se realizarán en total a las funciones `pueblo(A,PA)` y `consulta(A,B)`
  - $M$ , el largo de las cadenas.
- `pueblo(A, PA)`, que retorna la población **previa** del pueblo (0 si no existía). Sus parámetros son:
  - $A$ , la cadena de longitud  $M$  que identifica al pueblo.
  - $PA$ , la **nueva** población del pueblo.
- `consulta(A,B)`, que retorna la población total para esa consulta. Sus parámetros son:
  - $A$ , cadena de longitud  $M$  que indica el extremo izquierdo de la consulta.
  - $B$ , cadena de longitud  $M$  que indica el extremo derecho de la consulta.

### Evaluador local

El evaluador local lee de la entrada estándar:

- Una línea con la cantidad  $Q$  de llamados a las funciones `pueblo(A,PA)` y `consulta(A,B)`, y el largo  $M$  de los vectores. Realiza la llamada `inicializar(Q,M)`.
- Las  $Q$  consultas. Para cada una, lee una primera línea con un entero  $t = 1$  o  $t = 2$ , que indica el tipo de consulta.
- Si  $t = 1$ , lee en la siguiente línea la cadena  $A$ , de longitud  $M$ , y luego el entero  $P_A$ . Realiza la llamada `pueblo(A,PA)` y muestra el resultado por la salida estándar.
- Si  $t = 2$ , lee en las 2 siguientes líneas las cadenas  $A$  y  $B$ , de longitud  $M$ . Realiza la llamada `consulta(A,B)` y muestra el resultado por la salida estándar.

### Cotas

- $4 \leq Q \cdot M \leq 500.000$
- $0 \leq A_i, B_i \leq 9$
- En todo momento la población de cada pueblo es un entero entre 0 y  $10^{12}$  inclusive.
- En cada consulta,  $A$  siempre está estrictamente a la izquierda de  $B$ .

**Ejemplos**

Si el evaluador local recibe la siguiente entrada:

```

10 5
2
01234
01235
1
01234
1000
1
01235
1000
2
01233
01236
1
01233
500
1
01236
1000
1
01233
9500
1
01236
9000
2
00000
99999
2
99998
99999

```

Una implementación correcta deberá devolver:

```

0
0
0
2000
0
0
500
1000
20500
0

```

En cambio, si se recibe el siguiente ejemplo, correspondiente a la figura de la última hoja:

```

4 3
1
029
629
1
029
1000
1
977
100
2
029
989

```

Deberá devolver:

```

0
629
0
1100

```

**Subtareas**

1.  $Q \leq 100, M \leq 1.000$ . (5 puntos)
2.  $M \leq 6$ . (17 puntos)
3. No habrá llamadas a la función consulta. (10 puntos)
4. Todas las operaciones de tipo 1 ocurren antes que cualquiera de las operaciones de tipo 2. (18 puntos)
5. Las cadenas  $A, B$  únicamente contienen caracteres 0 y 1. (24 puntos)
6. Sin más restricciones. (26 puntos)