

Podando Pomelos

Contribución de Agustín Santiago Gutiérrez y Carlos Miguel Soto

Descripción del problema

Paul el Pomelista es encargado de mantener en buen estado el vasto jardín de pomelos de Linealandia. Linealandia tiene N pomelos, y se encuentran en una hilera numerados del 1 al N . Cada pomelo tiene un tiempo de poda $A[i]$ que es el tiempo que Paul tarda en podarlo, y además una complejidad de poda $B[i]$.

Algunos días, el presidente de Linealandia le pide a Paul que podo los pomelos en un rango $[L, R]$ de la hilera. Paul siempre comienza a podar primero el pomelo L , luego el $L + 1$, y así hasta que llega al pomelo R . La dificultad de podar el pomelo i es la complejidad de poda $B[i]$ más el tiempo de poda acumulado de podar ese pomelo y todos los anteriores que podó ese día, pues Paul se va cansando al podar pomelos. Para estar mentalmente preparado, a Paul le gustaría saber de antemano cuál será la dificultad máxima de podar un pomelo en el rango indicado. Formalmente, se quiere encontrar el máximo sobre todo $i \in [L, R]$ de $B[i] + A[L] + A[L + 1] + \dots + A[i]$.

Tenés que procesar **en orden** Q eventos, que pueden ser de tres tipos:

- $A \ i \ x$: $A[i]$, es decir tiempo de poda del pomelo i , cambia al nuevo valor x .
- $B \ i \ x$: $B[i]$, es decir la complejidad de poda del pomelo i , cambia al nuevo valor x .
- $C \ L \ R$: Paul **C**onsulta cuál es la dificultad máxima de podar un pomelo, al atender un pedido del presidente de podar los pomelos en el rango $[L, R]$.

Inicialmente, tanto el tiempo de poda $A[i]$ como la complejidad de poda $B[i]$ son todos 0.

Descripción de la función

Se debe implementar una función `pomelos(N, e, p1, p2)`. Sus parámetros son:

- N : La cantidad de pomelos
- $e, p1, p2$: Arreglos de Q elementos, que describen los eventos. El j -ésimo evento es de tipo $e[j]$ (cadena "A", "B" o "C").
 - Si $e[j] = "A"$ o $e[j] = "B"$, entonces $p1[j] = i$ y $p2[j] = x$ de acuerdo con la explicación anterior de los eventos.
 - Si $e[j] = "C"$ (es decir una consulta), entonces $p1[j] = L$ y $p2[j] = R$ que indican el rango $[L, R]$ de la consulta.

La función debe retornar un arreglo de enteros con las respuestas a todas las consultas (eventos de tipo C) en el mismo orden en que aparecen en `eventos`.

Evaluador local

El evaluador local lee de la entrada estándar con el siguiente formato:

- Dos enteros N y Q
- Q líneas, cada una con un evento de uno de los tres tipos explicados anteriormente.

Escribe en la salida los enteros del arreglo devuelto por la función, uno por línea.

Restricciones

- $1 \leq N, Q \leq 2 \cdot 10^5$
- $0 \leq x \leq 10^9$
- $1 \leq L \leq R \leq N$

Existe al menos una consulta (evento de tipo C).

Ejemplos

Si la entrada es:

2	5
A	1 1
A	2 1
B	2 3
C	1 2
C	1 1

La salida correcta es:

5
1

Subtareas

1. $N, Q \leq 50$ (5 puntos)
2. $N, Q \leq 2000$ (5 puntos)
3. $N, Q \leq 5 \cdot 10^4$ (20 puntos)
4. Ningún evento es de tipo A (6 puntos)
5. Ningún evento es de tipo B (7 puntos)
6. Todos los eventos de tipo A están al principio (11 puntos)
7. Todas las consultas están al final (17 puntos)
8. Sin restricciones adicionales (29 puntos)