

Invertir un segmento de una lista enlazada

En este ejercicio se trata de practicar las listas enlazadas dobles (utilizaremos las vistas en clase para implementar el TAD `deque`, aunque la nueva operación no sea una operación de colas dobles). Queremos una operación que dé la vuelta a un segmento (elementos en posiciones consecutivas) de la lista enlazada. Por ejemplo, si la lista está formada por los elementos 1 2 3 4 5 6 7 8, e invertimos el segmento que comienza en la posición 3 (las posiciones se numeran desde 1 hasta N , el número de elementos de la lista) y tiene longitud 4, entonces la lista resultante es 1 2 *6 5 4 3* 7 8 (donde se han marcado en cursiva los elementos invertidos).

Requisitos de implementación.

En la resolución del problema, se extenderá *mediante herencia* la clase `deque` con un método que invierta, de una lista genérica, un segmento especificado por su posición de comienzo dentro de la lista y su longitud.

El coste de la operación debe ser lineal con respecto al número de elementos en la lista. No pueden hacerse nuevos `news`; deben reutilizarse los nodos que ya existen en la lista enlazada.

También se añadirá un método que permita mostrar el contenido de la lista (sin modificarla) en una línea, separando sus elementos por espacios.

No modifiques ni subas al juez el fichero `deque_eda.h` cuya clase `deque` debes extender.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso se muestra en dos líneas. La primera contiene tres números: N , el número de elementos de la lista (un número entre 1 y 100.000); P , la posición de comienzo del segmento (un número entre 1 y N); y $L \geq 1$, la longitud (número de elementos) del segmento. Se garantiza que el segmento está incluido en la lista, es decir, $P + L - 1 \leq N$. En la segunda línea se muestran los N elementos de la lista, números entre 1 y 1.000.000.

Salida

Para cada caso de prueba se escribirá en una línea la lista después de invertir el segmento indicado.

Entrada de ejemplo

```
8 3 4
1 2 3 4 5 6 7 8
8 1 8
1 2 3 4 5 6 7 8
8 3 1
1 2 3 4 5 6 7 8
```

Salida de ejemplo

```
1 2 6 5 4 3 7 8
8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8
```

Autor: Facultad Informática (UCM)