

Universidad Nacional Autónoma de México
Facultad de Ciencias
Estructuras de Datos 2022-2
Tarea 02: Polinomio de redireccionamiento y Listas

Pedro Ulises Cervantes González Yessica Janeth Pablo Martínez,
confundeme@ciencias.unam.mx yessica_j_pablo@ciencias.unam.mx

Jorge Macías Gómez
jorgemacias@ciencias.unam.mx

Fecha de entrega: 22 de Marzo del 2022
Hora límite de entrega: 23:59

1. Polinomio de redireccionamiento

.- Dado el siguiente arreglo A escrito en Java de dimensiones $3 \times 2 \times 3$, ubicada en la posición D, donde cada entrada mide k bytes, obtener a los elementos mostrando paso a paso de la dirección (así como lo vimos en clase):

- A[1][0][2]
- A[2][1][2]
- A[1][1][2]
- A[0][1][2]

```
int A[][][] = {  
    { {a,b,c},  
      {d,e,f},  
      {}},  
    { {g,h,i},  
      {j,k,l},  
      {}},  
    { {m,n,o},  
      {p,q,r},  
      {}}  
}
```

2. Listas

.-Dada la siguiente lista simple : {2,4,6,8,10,12}, realiza las siguientes operaciones en ese mismo orden con dibujos paso a paso y regresa la lista final (así como lo vimos en clase):

- Insertar en la posición 3 al elemento -1
- Elimina al elemento de la posición 2

- Regresa el tamaño de la lista
- Inserta en la posición 2 al elemento 3.3
- Obtener al elemento de la posición 2

.- Dada una lista doblemente ligada : {1,3,5,7,9,11,13}, realiza las siguientes operaciones en ese mismo orden con dibujos paso a paso y regresa la lista final (así como lo vimos en clase):

- Insertar en la posición 2 al elemento 0
- Insertar en la posición 0 al elemento 6
- Elimina al elemento de la posición 0
- Regresa el tamaño de la lista
- Obtener al elemento de la posición 2

Respuestas

Equipo:

Bonilla Reyes Dafne
García Ponce José Camilo

- Ejercicio 1

Primero veamos quienes son d , n_1 , n_2 y n_3 :

$$\begin{aligned}d &= 3 \\n_1 &= 3, n_2 = 2 \text{ y } n_3 = 3\end{aligned}$$

- $A[1][0][2]$

Observemos quienes son i_1 , i_2 y i_3

$$i_1 = 1, i_2 = 0 \text{ y } i_3 = 2$$

Entonces:

$$\begin{aligned}A[1][0][2] &= D + (1 * 2 * 3 + 0 * 3 + 2)k \\A[1][0][2] &= D + (6 + 0 + 2)k \\A[1][0][2] &= D + (8)k \\A[1][0][2] &= D + 8k\end{aligned}$$

Por lo tanto, nos desplazamos 8 lugares y obtenemos que:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r



$$A[1][0][2] = i$$

- $A[2][1][2]$

Observemos quienes son i_1 , i_2 y i_3

$$i_1 = 2, i_2 = 1 \text{ y } i_3 = 2$$

Entonces:

$$A[2][1][2] = D + (2 * 2 * 3 + 1 * 3 + 2)k$$

$$A[2][1][2] = D + (12 + 3 + 2)k$$

$$A[2][1][2] = D + (17)k$$

$$A[2][1][2] = D + 17k$$

Por lo tanto, nos desplazamos 17 lugares y obtenemos que:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r



$$A[2][1][2] = r$$

- $A[1][1][2]$

Observemos quienes son i_1 , i_2 y i_3

$$i_1 = 1, i_2 = 1 \text{ y } i_3 = 2$$

Entonces:

$$A[1][1][2] = D + (1 * 2 * 3 + 1 * 3 + 2)k$$

$$A[1][1][2] = D + (6 + 3 + 2)k$$

$$A[1][1][2] = D + (11)k$$

$$A[1][1][2] = D + 11k$$

Por lo tanto, nos desplazamos 11 lugares y obtenemos que:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r



$$A[1][1][2] = l$$

- $A[0][1][2]$

Observemos quienes son i_1 , i_2 y i_3

$$i_1 = 0, i_2 = 1 \text{ y } i_3 = 2$$

Entonces:

$$A[0][1][2] = D + (0 * 2 * 3 + 1 * 3 + 2)k$$

$$A[0][1][2] = D + (0 + 3 + 2)k$$

$$A[0][1][2] = D + (5)k$$

$$A[0][1][2] = D + 5k$$

Por lo tanto, nos desplazamos 5 lugares y obtenemos que:

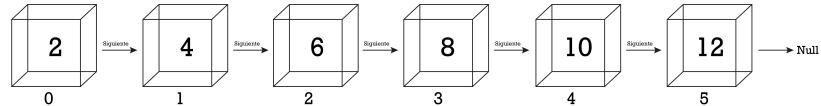


$$A[0][1][2] = f$$

■ Ejercicio 2

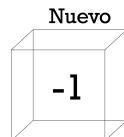
1. Lista 1

Lista simplemente ligada: {2, 4, 6, 8, 10, 12}

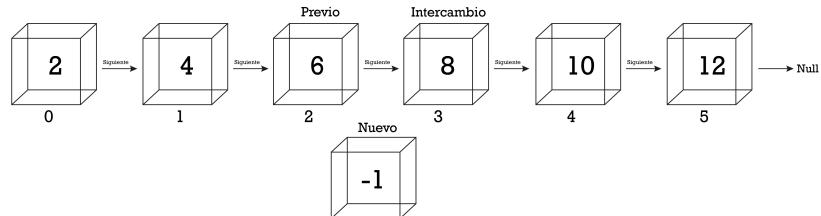


- Paso 1: Insertar en la posición 3 al elemento -1

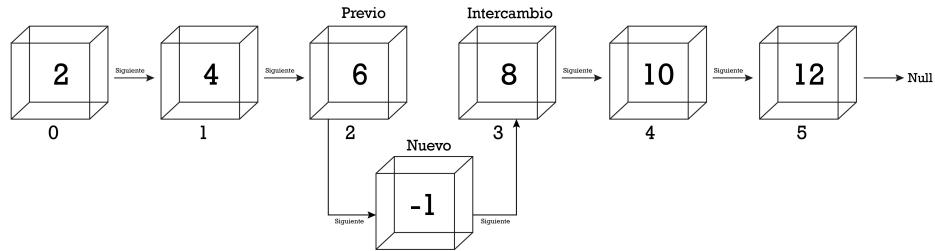
- Creamos un nuevo nodo llamado *nuevo*, el cual contiene el elemento por agregar.



- Iteramos, empezando por la cabeza, hasta encontrar el nodo con el índice que queremos y al nodo anterior a este, llamándolos *intercambio* y *previo* respectivamente.

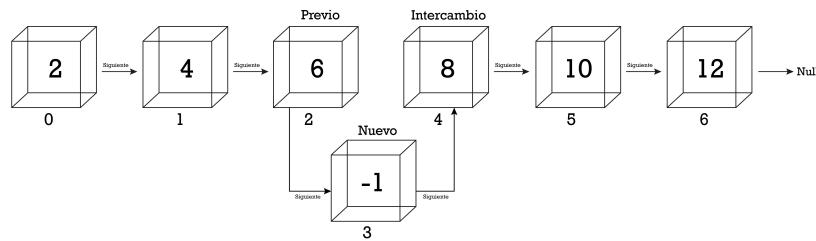


- c) Como *intercambio* no es la cabeza ni el último, hacemos los siguientes cambios de referencias:
 previo.siguiente = nuevo
 nuevo.siguiente = intercambio



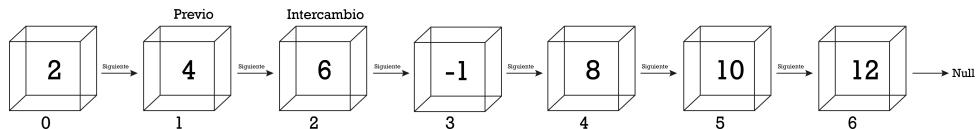
d) Aumentamos en 1 la variable longitud de la lista.

e) Actualizamos los índices de los nodos.

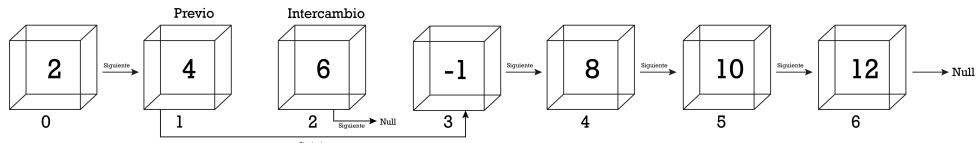


- Paso 2: Elimina al elemento de la posición 2

- a) Iteramos, empezando por la cabeza, hasta encontrar el nodo con el índice que queremos y al nodo anterior a este, llamándolos *intercambio* y *previo* respectivamente.

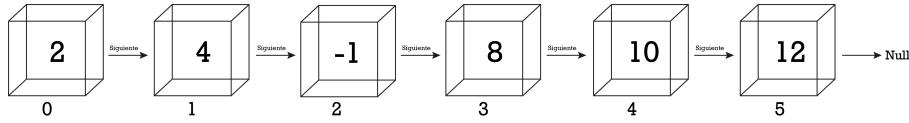


- b) Como *intercambio* no es la cabeza ni el último, hacemos los siguientes cambios de referencias:
 previo.siguiente = intercambio.siguiente
 intercambio.siguiente = null
 Y el recogedor de basura se encargará del nodo *intercambio*, ya que es inalcanzable.



- c) Disminuimos en 1 la variable longitud de la lista.

d) Actualizamos los índices de los nodos.



- Paso 3: Regresa el tamaño de la lista

a) Regresamos la variable longitud de la lista.

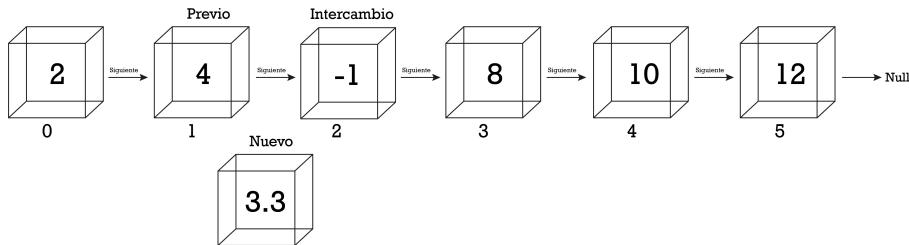


- Paso 4: Insertar en la posición 2 al elemento 3.3

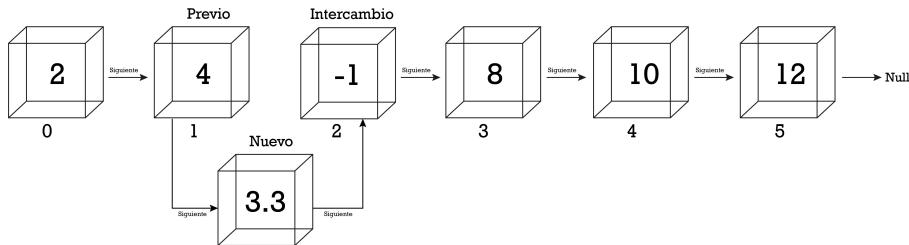
a) Creamos un nuevo nodo llamado *nuevo*, el cual contiene el elemento por agregar.



b) Iteramos, empezando por la cabeza, hasta encontrar el nodo con el índice que queremos y al nodo anterior a este, llamándolos *intercambio* y *previo* respectivamente.

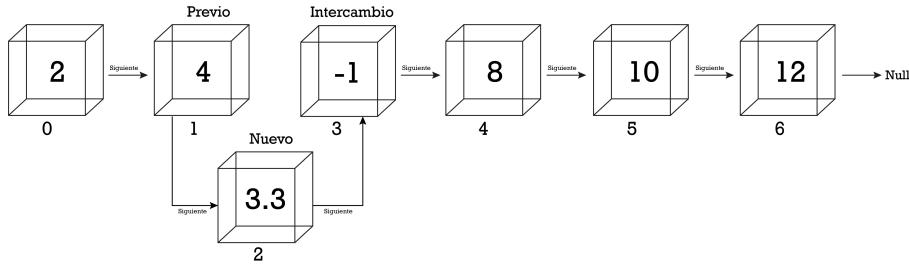


c) Como *intercambio* no es la cabeza ni el último, hacemos los siguientes cambios de referencias
`previo.siguiente = nuevo`
`nuevo.siguiente = intercambio`



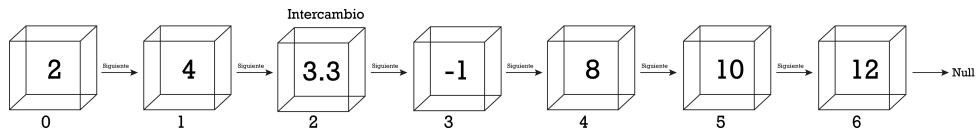
d) Aumentamos en 1 la variable longitud de la lista.

e) Actualizamos los índices de los nodos.

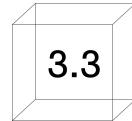


- Paso 5: Obtener al elemento de la posición 2

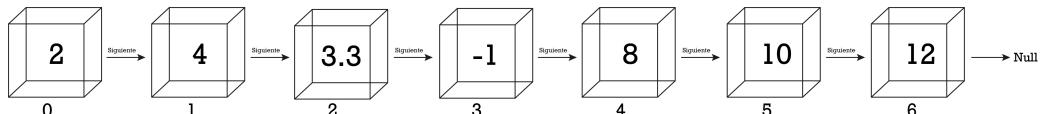
a) Iteramos, empezando por la cabeza, hasta encontrar el nodo con el índice que queremos, llamándolo *intercambio*.



- b) Devolvemos el elemento de *intercambio*.

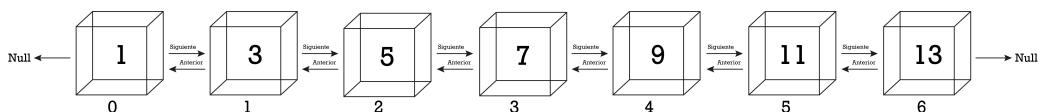


- Paso 6: Devolvemos la lista.



2. Lista 2

Lista doblemente ligada: {1, 3, 5, 7, 9, 11, 13}

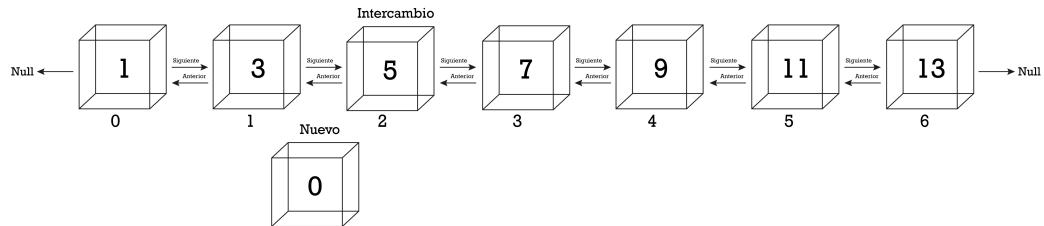


- Paso 1: Insertar en la posición 2 al elemento 0

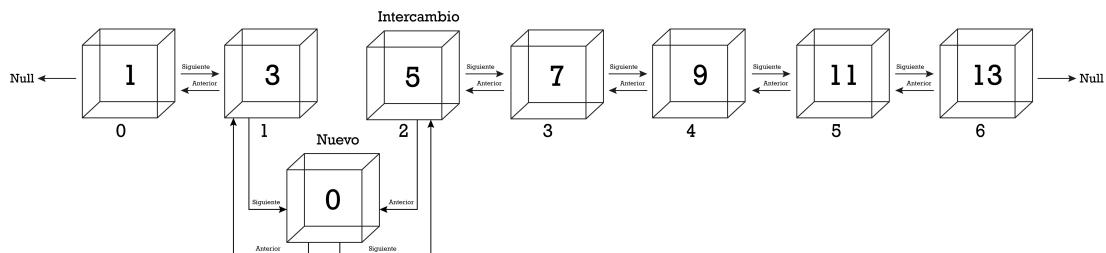
a) Creamos un nuevo nodo llamado *nuevo*, el cual contiene el elemento por agregar.



b) Iteramos, empezando por la cabeza, hasta encontrar el nodo con el índice que queremos, llamándolo *intercambio*.

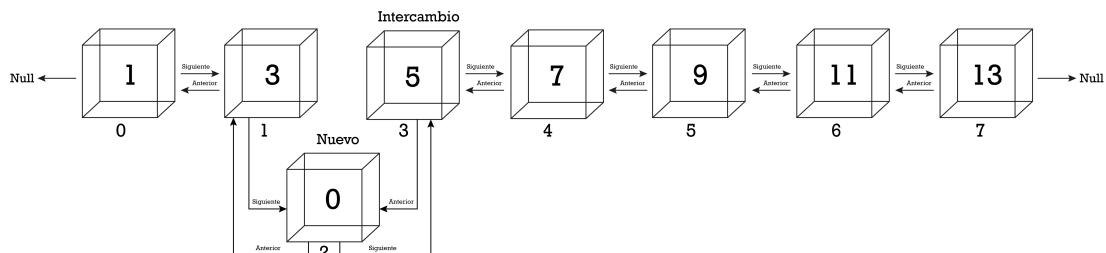


c) Como *intercambio* no es la cabeza ni el último, hacemos los siguientes cambios de referencias:
intercambio.anterior.siguiente = nuevo;
nuevo.anterior = intercambio.anterior;
intercambio.anterior = nuevo;
nuevo.siguiente = intercambio;



d) Aumentamos en 1 la variable longitud de la lista.

e) Actualizamos los índices de los nodos.

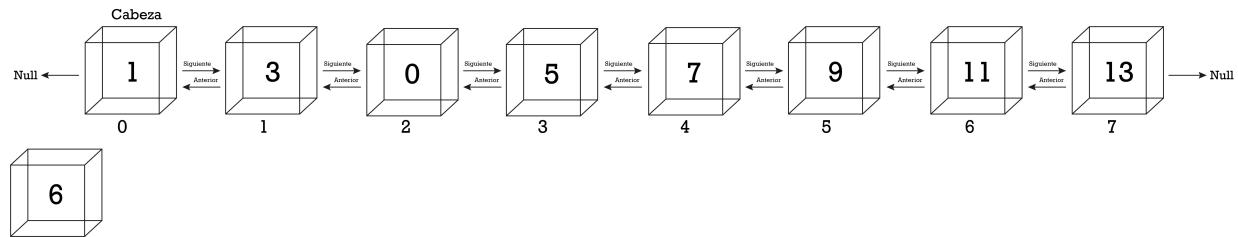


- Paso 2: Insertar en la posición 0 al elemento 6

a) Creamos un nuevo nodo llamado *nuevo*, el cual contiene el elemento por agregar.

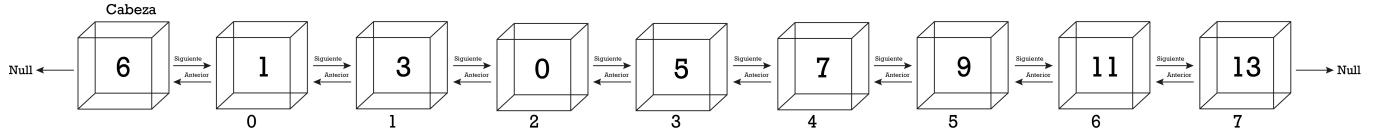


b) Como la posición 0 es la cabeza, no iteramos.



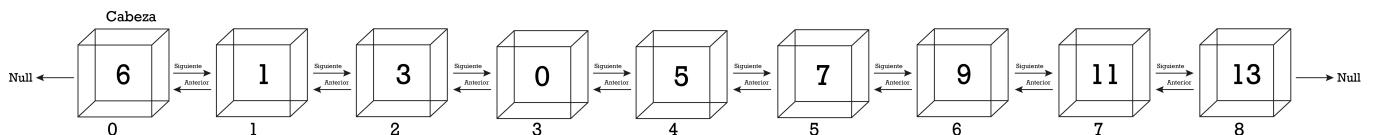
c) Como queremos agregar al inicio, hacemos los siguientes cambios de referencias:

cabeza.anterior = nuevo
nuevo.siguiente = cabeza
cabeza = nuevo



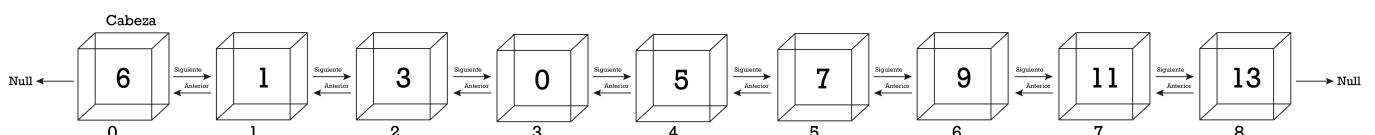
d) Aumentamos en 1 la variable longitud de la lista.

e) Actualizamos los índices de los nodos.

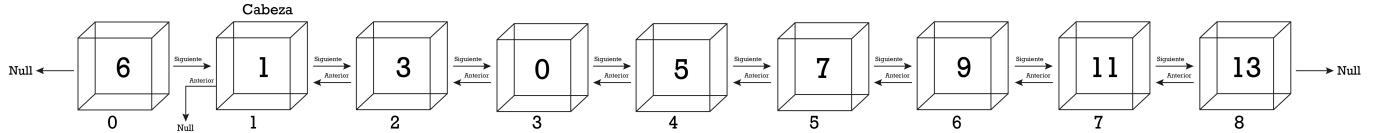


- Paso 3: Elimina al elemento de la posición 0

a) Como la posición 0 es la cabeza no iteramos.

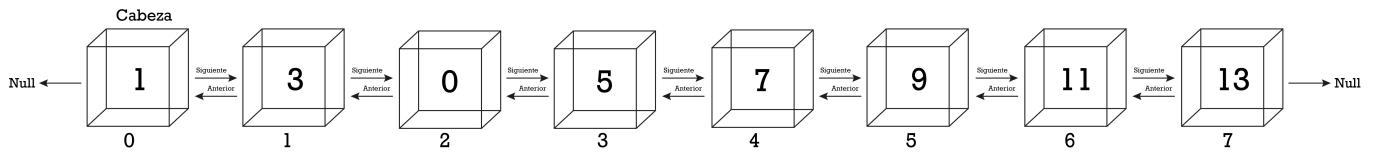


- b) Como queremos eliminar al inicio, hacemos los siguientes cambios de referencias:
`cabeza.siguiente.anterior = null`
`cabeza = cabeza.siguiente`
Y el recogedor de basura se encargará del nodo que antes era la cabeza, ya que es inalcanzable.



c) Disminuimos en 1 la variable longitud de la lista.

d) Actualizamos los índices de los nodos.



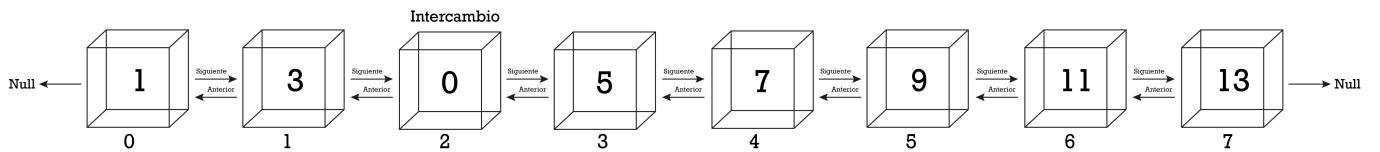
- Paso 4: Regresa el tamaño de la lista

a) Regresamos la variable longitud de la lista.

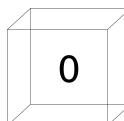


- Paso 5: Obtener al elemento de la posición 2

a) Iteramos, empezando por la cabeza, hasta encontrar el nodo con el índice que queremos, llamándolo *intercambio*.



b) Devolvemos el elemento de *intercambio*.



- Paso 6: Devolvemos la lista.

