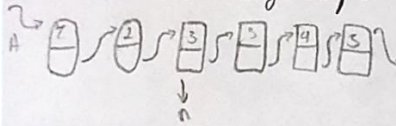


Díaz Hernández Marcos Bryan

Ejercicio Virtual 11

1) Escribe un código genérico, para añadir un nuevo nodo en cualquier posición de una lista ligada simple.

- No es simple que el código valide la existencia de nodos suficientes.
- Si el usuario ingresa posición 1, se deberá añadir al principio.

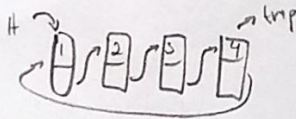
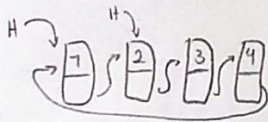


```

1) Nodo tmp = head;
2) int a;
3) indica posición → guarda en a
4) for (i = 1; i < a; i++) {
    tmp = tmp.next; → guarda n
5) Nodo nuevo.info = 10;
6) Nodo aux = head;
7) for (i = 1; i < a - 1; i++) {
8)     aux = aux.next;
9) nuevo.next = tmp;
10) aux = nuevo;

```

2) Escribe el código de la eliminación del primer y último nodo de una lista ligada circular.



eliminar primero

```

1) Nodo tmp = Head
2) while (tmp.next != Head) {
3)     tmp = tmp.next;
4) Head = head.next;
5) tmp.next = Head;
6) Nodo delete = head;
   while (delete.next != Head) {
       delete = tmp.next;
7) delete = null;

```

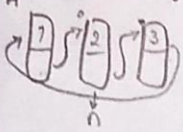
eliminar último

```

1) Nodo tmp = Head
2) while (tmp.next != Head) {
    tmp = tmp.next;
3) tmp = null;
4) Nodo aux = Head;
5) while (aux.next != Head) {
    aux = aux.next;
6) aux.next = Head;

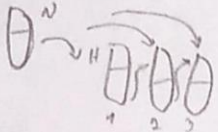
```

3) Escribe un código de como se puede hacer una inserción en una lista circular sin desplazar "head" y sin usar un nodo "tmp".



1) Nodo new = Head  
2) while (new.next != Head)  
new

while (Head.next != Head)  
Head.next = Head.next.next  
Head.next = new → pierde



1) Decidir la posición  $n=2$   
2) Nodo new = Head  
if ( $n > \text{nodes}$ ) { } Al final  
Head.next.next...n = new

4) if ( $n=7$ )  
Nodo new = Head  
Head.next.next...n = new.

• Para ser el final y primero se necesita repetir

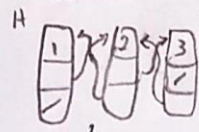
Head.next.next... n veces para establecer la dirección de memoria anterior

3) if ( $n = \text{nodes}$ ) { } Al mitad o n parte  
for ( $i=1$ ;  $i \leq n$ ;  $i++$ ) { }  
new = new.next  
Head.next.next... n-1 = new

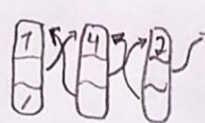
• Para ubicar las direcciones del anterior y del siguiente del nuevo se busca iniciar por el head para buscar las direcciones del siguiente y que apunte al  $n+1$ .

Lo más confuso es el como repetir el Head.next sin que se pierdan las direcciones de memoria.

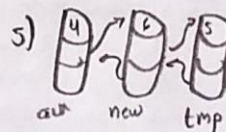
4) Escribe el código para añadir un nuevo nodo en la segunda posición de una lista ligada doble.



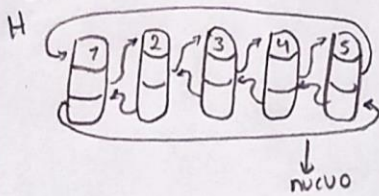
$n=1$   $\hookrightarrow n=2$



1) Nodo tmp = head  
2) tmp = head.next  
3) Nodo nuevo  
4) head.next = nuevo  
5) nuevo.next = tmp  
6) nuevo.prev = head  
7) tmp.prev = nuevo



5) Escribe el código para añadir un nuevo nodo entre nodo 4 y el nodo 5 de la lista ligada circular doble mostrada.



1) Nodo tmp = head  
2) while (tmp.next != head)  
tmp = tmp.next  
3) Nodo aux = tmp.prev  
4) Nodo new  
5) new.next = tmp  
6) new.prev = aux  
7) aux.next = new  
8) tmp.prev = new