Universidad Tecnológica del Norte de Guanajuato

Ingeniería en Tecnologías de la Información e Innovación Digital: Especialización en Desarrollo de Software y Multiplataforma

Valeria García Gaona (1224100671)

GTID141

Estructura de datos - VisuAlgo

Gabriel Barron

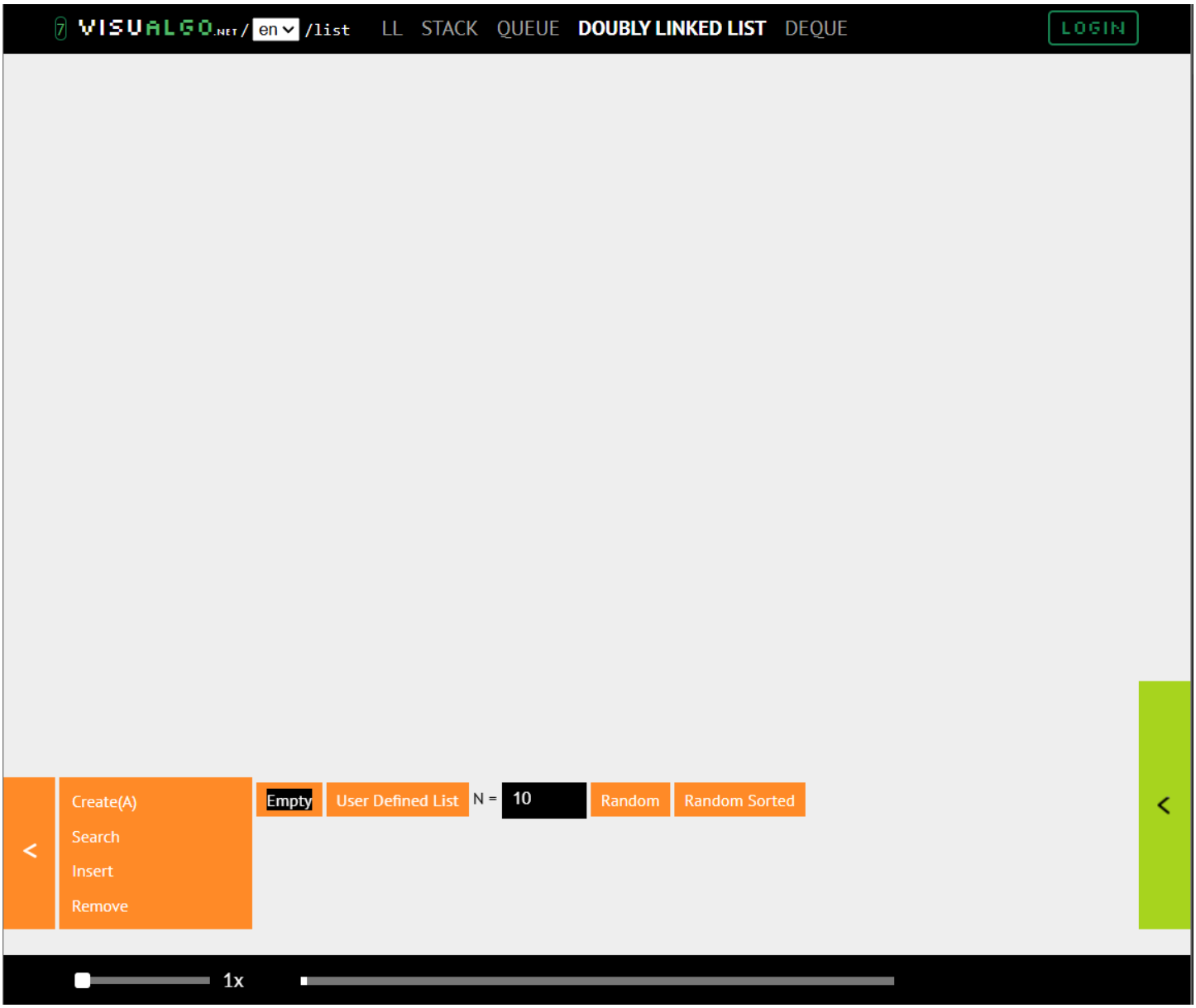
Dolores Hidalgo Gto. C.I.N.

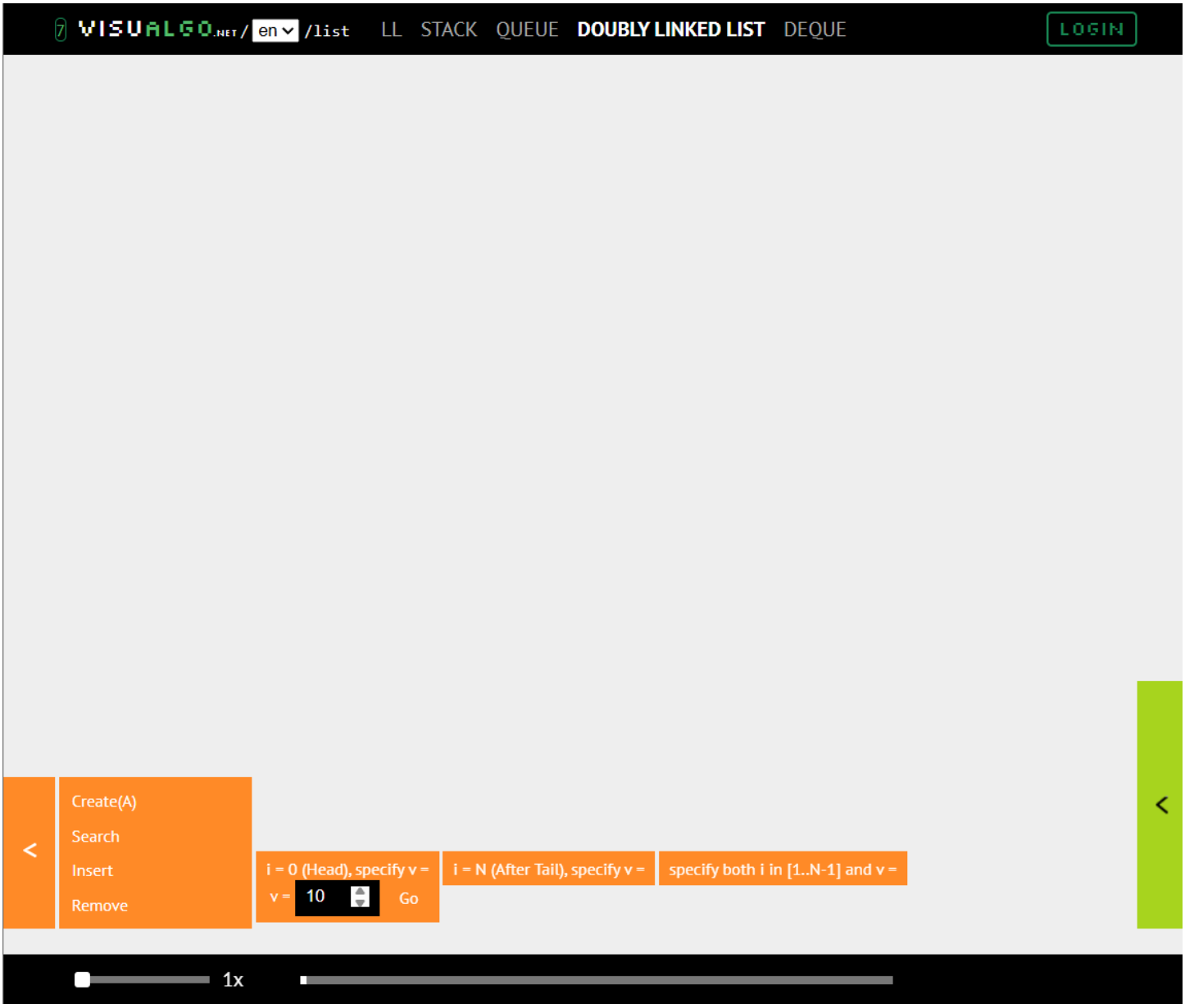
VISUALGO.NET / en / list LL STACK QUEUE **DOUBLY LINKED LIST** DEQUE LOGIN

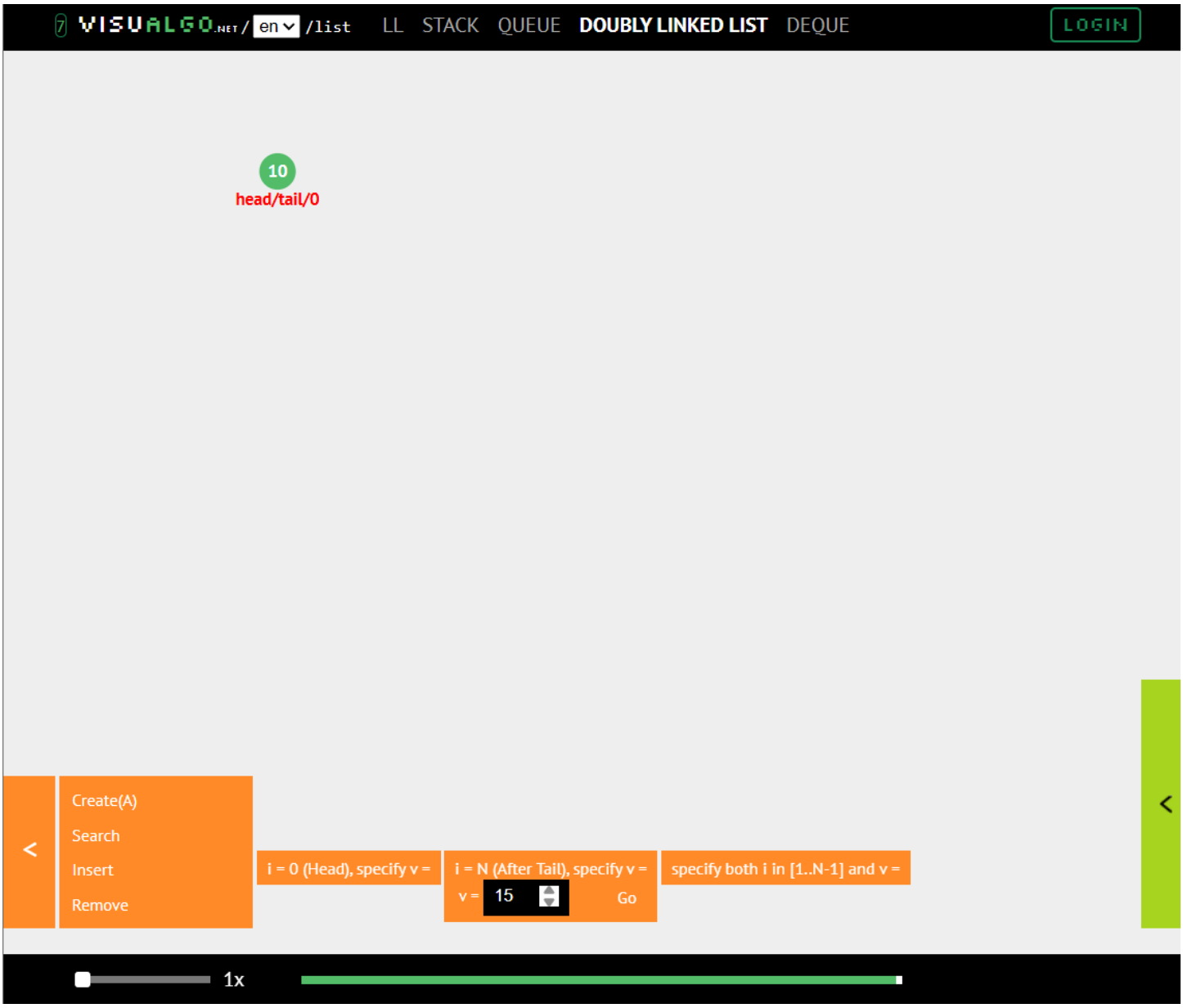
Create(A)
Search
Insert
Remove

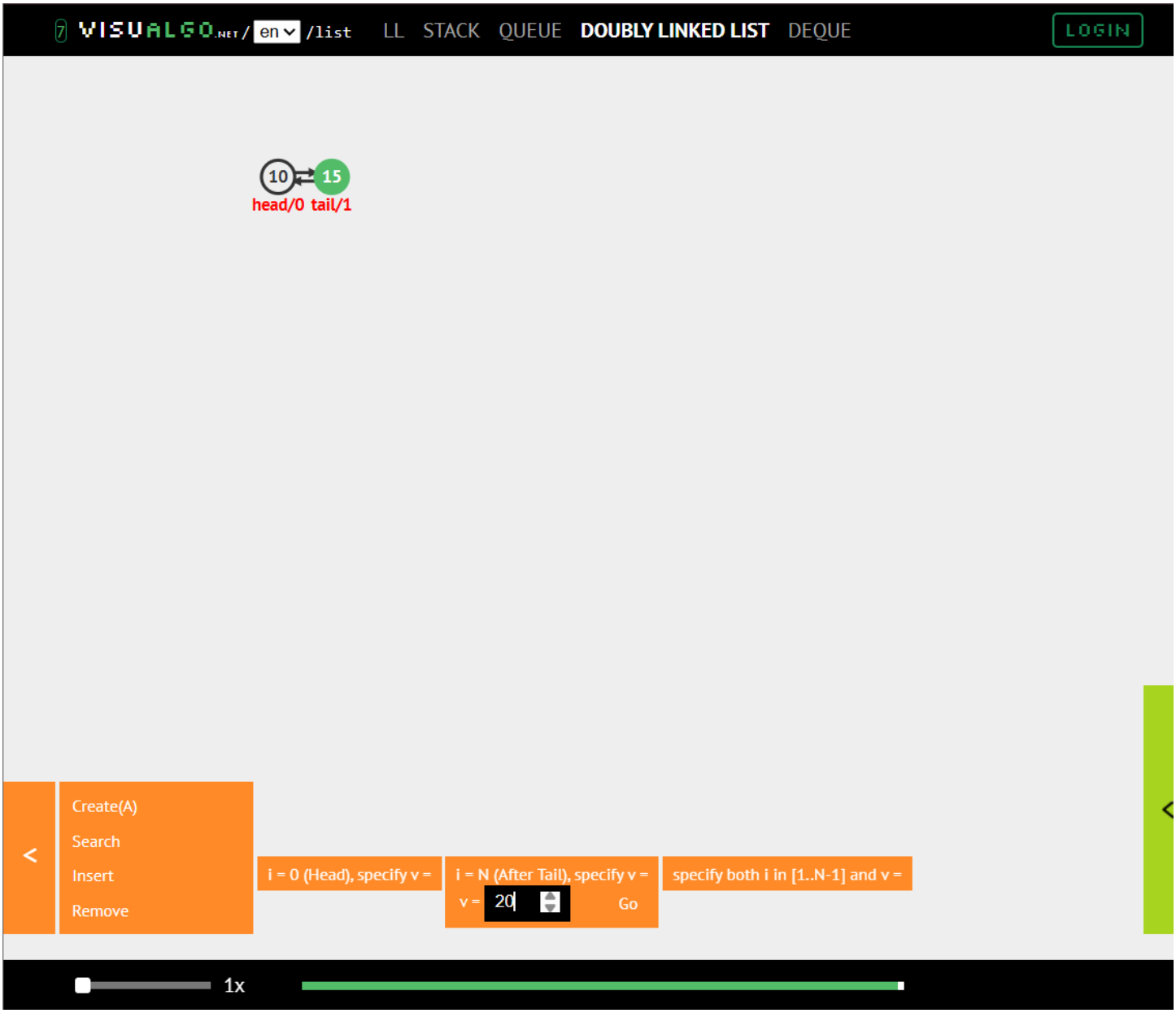
Empty | User Defined List | N = 10 | Random | Random Sorted

1x


VISUALGO.NET / en /list
LL
STACK
QUEUE
DOUBLY LINKED LIST
DEQUE
LOGIN
Create(A)
Search
Insert
Remove
Empty
User Defined List
N =
10
Random
Random Sorted
1x


VISUALGO.NET/ en /list
LL
STACK
QUEUE
DOUBLY LINKED LIST
DEQUE
LOGIN
<
Create(A)
Search
Insert
Remove
i = 0 (Head), specify v =
v = 10
Go
i = N (After Tail), specify v =
specify both i in [1..N-1] and v =
<
1x


VISUALGO.NET / en /list
LL
STACK
QUEUE
DOUBLY LINKED LIST
DEQUE
LOGIN
10
head/tail/0
Create(A)
Search
Insert
Remove
i = 0 (Head), specify v =
i = N (After Tail), specify v =
v = 15
Go
specify both i in [1..N-1] and v =
1x


VISUALGO.NET / en /list
LL
STACK
QUEUE
DOUBLY LINKED LIST
DEQUE
LOGIN
10
15
head/0 tail/1
Create(A)
Search
Insert
Remove
i = 0 (Head), specify v =
i = N (After Tail), specify v =
v = 20
Go
specify both i in [1..N-1] and v =
1x

Create(A)
Search
Insert
Remove

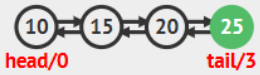
i = 0 (Head), specify v =

i = N (After Tail), specify v =
v = 25
Go

specify both i in [1..N-1] and v =

1x

VISUALGO.NET / en / list LL STACK QUEUE **DOUBLY LINKED LIST** DEQUE LOGIN



<

Create(A)

Search

Insert

Remove

i = 0 (Head), specify v =

i = N (After Tail), specify v =

v = 30 Go

specify both i in [1..N-1] and v =

<

1x

VISUALGO.NET / en /list LL STACK QUEUE **DOUBLY LINKED LIST** DEQUE LOGIN

Insert 30 at tail

tail points to vtx.
The whole operation is O(1) if we maintain the tail pointer.

>

1x

VISUALGO.NET / en / list LL STACK QUEUE **DOUBLY LINKED LIST**

<

Create(A)

Search

Insert

Remove

Remove i = 0 (Head)

Remove i = N-1 (Tail)

specify i in [1..N-2]

i = 3 Go

1x

VISUALGO.NET / en /list LL STACK QUEUE **DOUBLY LINKED LIST** DEQUE LOGIN

Remove index 3

Re-layout the Linked List for visualization (not in the actual Linked List). The whole process is still O(N).

1x

VISUALGO.NET / en / list LL STACK QUEUE **DOUBLY LINKED LIST** DEQUE LOGIN

<

Create(A)

Search

Insert

Remove

v = 30 Go

<

1x

Search 30

Comparing 20 (index = 2) with v = 30.
20 is not equal to 30 so we have to continue.

VISUALGO.NET / en /list LL STACK QUEUE **DOUBLY LINKED LIST** DEQUE LOGIN

Search 30

Found value v = 30 at this highlighted vertex so we return index 3.
The whole operation is O(N).

1x

1. ¿Qué sucede con los punteros cuando se inserta o elimina un nodo?

Cuando se inserta al inicio: cabeza pasa a ser segundo nodo y el nuevo nodo apunta a la cabeza anterior. En medio: se actualiza el nodo al nuevo nodo y el nuevo nodo al que sostenía el anterior. Al final: el último nodo apunta al nuevo y el nuevo es null.

Cuando se elimina al inicio: cabeza pasa a ser segundo nodo. En medio: se actualiza el nodo anterior al siguiente del eliminado. Al final: el penúltimo nodo se actualiza a null.

2. ¿Cómo afecta la posición de un nodo (inicio, medio, final) al tiempo de búsqueda?

Inicio: tiempo constante y acceso inmediato

Medio: tiempo lineal, se tiene que recorrer la mitad de la lista

Final: tiempo lineal, se recorre toda la lista

3. ¿Qué ventajas tiene recorrer una lista enlazada frente a otras estructuras como arreglos?

Para la inserción y eliminación funcional no requiere mover elementos, su tamaño es dinámico, tiene un uso de memoria eficiente

4. ¿Cómo podrías implementar la comprobación de una lista vacía en un lenguaje de programación como Java?

```
public class Nodo {
    int dato;
    Nodo siguiente;
}
public class ListaEnlazada {
    Nodo cabeza;

    public boolean estaVacia() {
        return cabeza == null;
    }
}
```