UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica etsinf

Tema 5. Tipos de Datos Lineales

Programación (PRG)

Jorge González Mollá

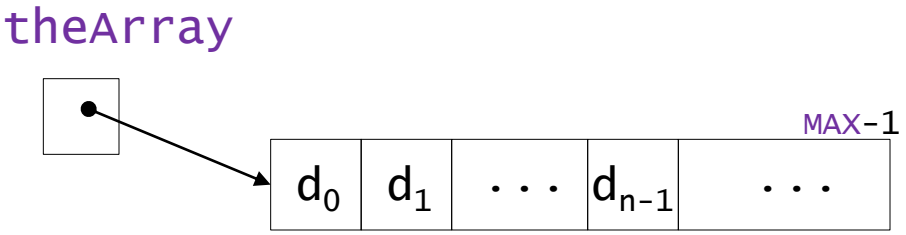
Departamento de Sistemas Informáticos y Computación

Índice

Secuencias con Arrays

Representación con arrays

- Una manera de representar una secuencia de datos del mismo tipo es poniendo sus elementos en las sucesivas posiciones de un array (con una longitud tal que permita almacenar la secuencia completa).

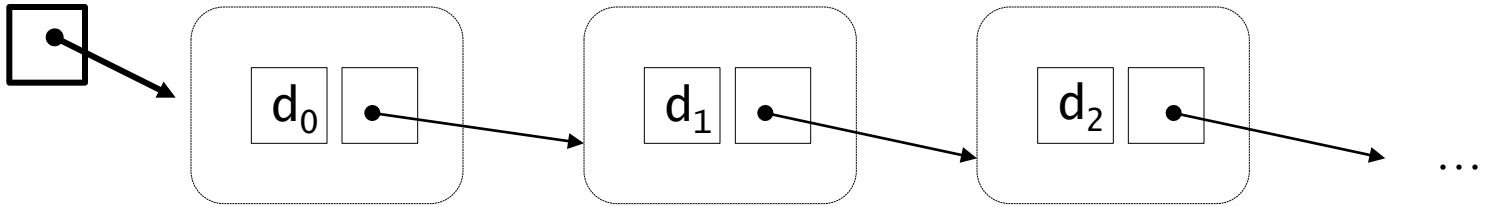


- Proporciona un acceso directo a cualquier elemento de la secuencia, es decir, con un coste constante independientemente de su tamaño.
- La talla de la secuencia está delimitada por la longitud `MAX` del array.
- La inserción/eliminación del dato de la posición `i`-ésima del array exige desplazar los datos del subarray derecho `theArray[i..n-1]`.

Secuencias de Nodos Enlazados

Representación enlazada

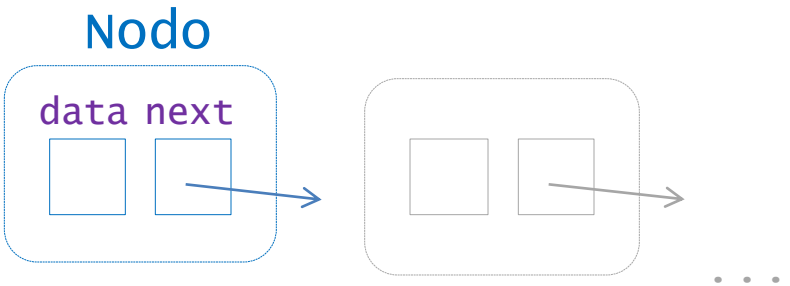
- Las secuencias de datos del mismo tipo se gestionan a la demanda, es decir, el espacio de memoria reservada para almacenar los datos es dinámico y (de)crece según metas/saques datos de la secuencia.
- Todo dato de la secuencia tiene asociado un enlace o referencia, que nos permite el acceso al siguiente dato dentro de la secuencia.



- El acceso a los datos de la secuencia ya no es directo como antes, sino por su posición en la misma, i.e. coste lineal respecto a la talla.

Atributos

- *Nodo:* Objeto que encapsula un dato
  y el enlace al siguiente objeto *Nodo*




```
/**
 * Clase NodeInt: nodos cuyo
 * dato es de tipo int
 */
public class NodeInt {

   int data;
   NodeInt next;


   ...
}
```

Constructores

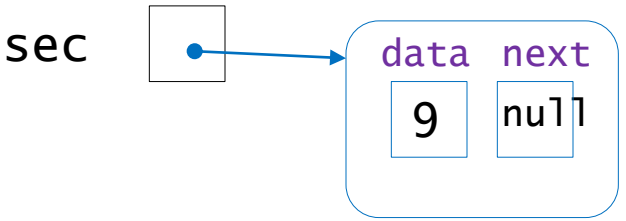
En la clase se van a implementar dos tipos de constructores:

- Parcial: Crea un nodo con un dato d que no tiene siguiente.

```
public class NodeInt {

   int data;
   NodeInt next;

   /** Constructor Parcial */
   NodeInt (int d) {
      data = d;
      next = null;
   }
    ...
}
```

Ejemplo:

```
NodeInt sec;
sec = new NodeInt(9);
```

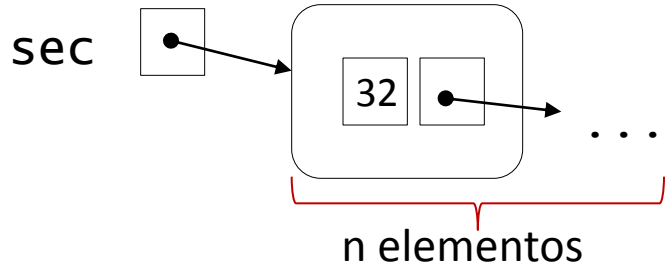


Constructores

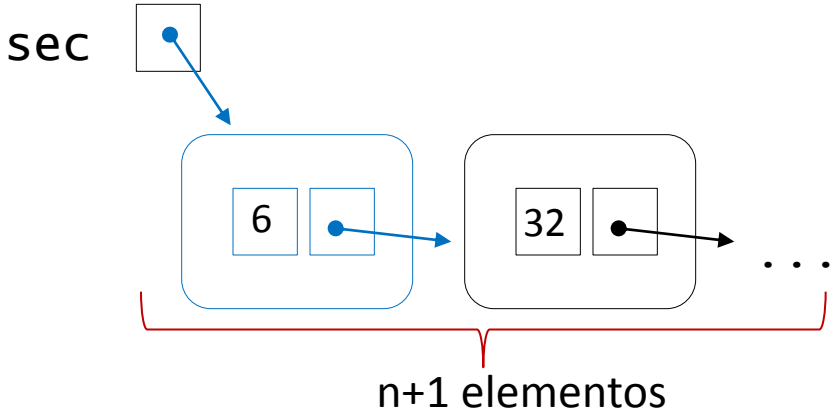
- Total: Crea un nodo con un dato d enlazado a un nodo preexistente.

```
public class NodeInt {

   int  data;
   NodeInt next;

   /** Constructor Parcial */
   NodeInt (int d) {
      data = d;
      next = null;
    // this(d, null);
   }

   /** Constructor Total */
   NodeInt (int d, NodeInt s) {
      data = d;
      next = s;
   }
}
```

Ejemplo:

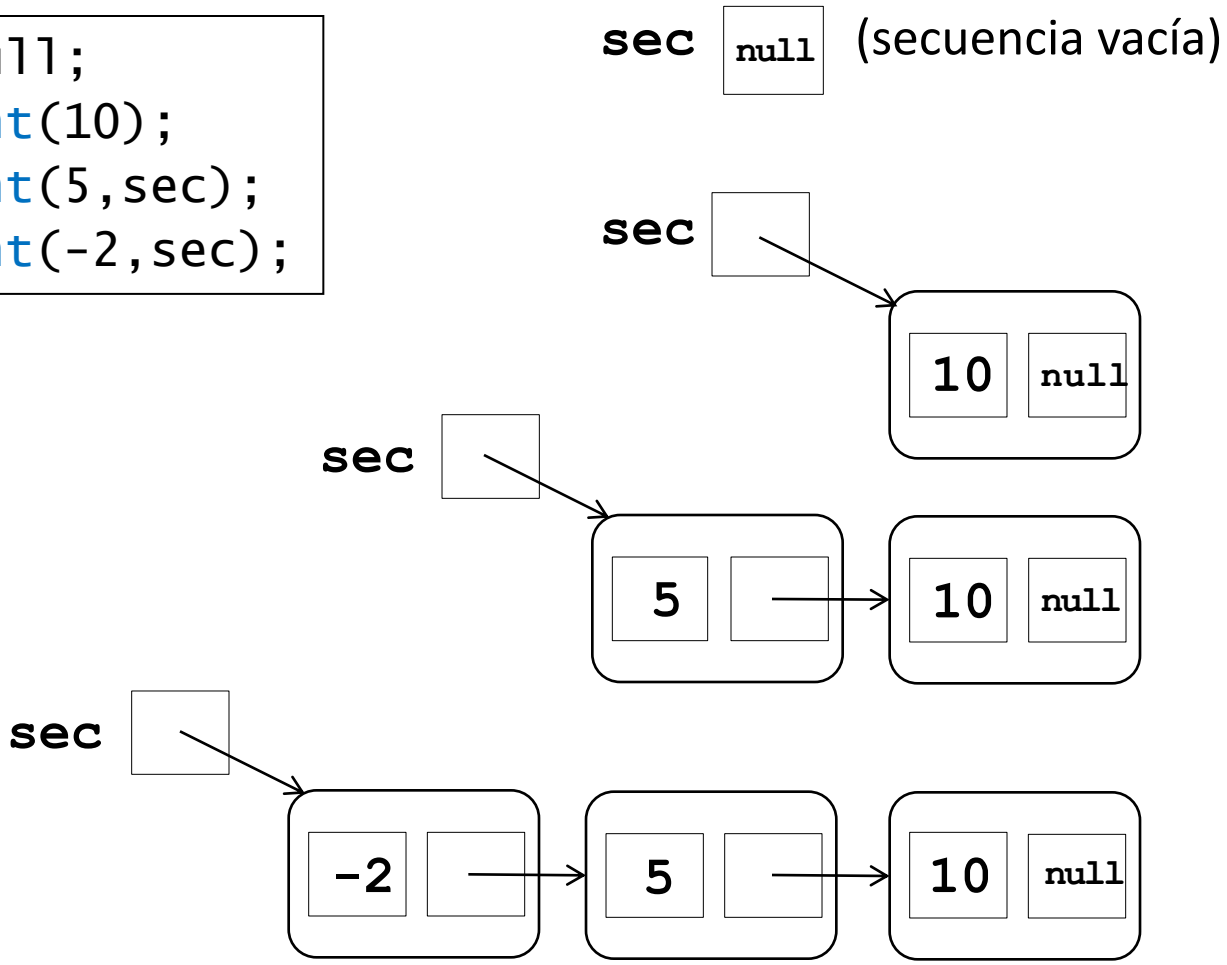



```
sec = new NodeInt(6,sec);
```



Ejemplo: Inserción delantera

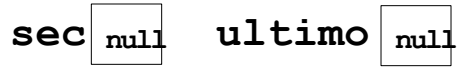
```
NodeInt sec = null;
sec = new NodeInt(10);
sec = new NodeInt(5,sec);
sec = new NodeInt(-2,sec);
```



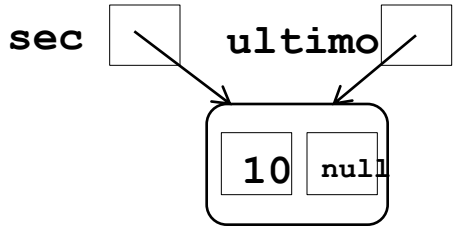
Ejemplo: Inserción trasera

- Los constructores de `NodeInt` facilitan la inserción en cabeza.
- La manipulación explícita de enlaces permite acceder a otras posiciones.

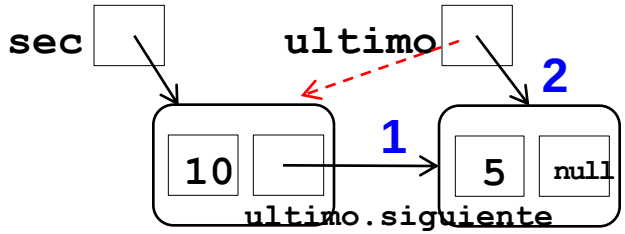
```
NodeInt sec = null, ultimo = null;
```




```
sec = new NodeInt(10);
ultimo = sec;
```




```
ultimo.siguiente = new NodeInt(5); // 1
ultimo = ultimo.siguiente; // 2
```




```
ultimo.siguiente = new NodeInt(-2);
ultimo = ultimo.siguiente;
```