

ESTRUCTURAS DE DATOS Y ALGORITMOS II

CLASE 1

ESTRUCTURAS DE DATOS Y ALGORITMOS II

CLASE 1

¿POR QUÉ?

PROBLEMA

A photograph of an open closet door showing a chaotic arrangement of clothing items. On the left, a wooden hanger holds a pink shirt and a red belt. Below it, a blue t-shirt hangs. In the center, a green dress and a patterned top are visible. To the right, a denim jacket and a red cap are hanging. A large yellow garment is draped over the right side. At the bottom left, several pairs of shoes are stacked, including white lace-ups and a black sandal. A straw hat sits on top of the shoes. The background shows more clothes hanging on the back wall.

EL PLACARD

EL PLACARD

¿qué pasa si tenemos sólo un estante, laaaaaargo, pero que no entra mas de una prenda de alto ?

¿y si la puerta es muy pequeña, y adentro hay un perchero giratorio?

condicionantes, requerimientos

¿QUÉ ES UNA ESTRUCTURA DE DATOS?

¿Qué es una estructura de datos?

“Una estructura de datos es una forma particular de organizar la información en una computadora de manera tal que pueda ser usada eficientemente.”

Paul E. Black - "Dictionary of Algorithms and Data Structures"

“Una estructura de datos es una forma de almacenar y organizar información para facilitar el acceso y las modificaciones.”

*[Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein] -
"Introduction to algorithms"*

Propiedades Reglas Restricciones

¿Qué es un algoritmo?

“un algoritmo es un procedimiento computacional bien definido que toma un valor/es como entrada y produce un valor/es como salida. Es por lo tanto una secuencia de pasos que transforman una entrada en una salida.”

“También puede verse como una herramienta para resolver un problema computacional bien especificado.”

¿POR QUÉ?

“frecuentemente tendremos que identificar y balancear las ventajas y desventajas acerca de una solución. Como científicos de la computación, además de nuestra habilidad de resolver problemas, también necesitaremos conocer herramientas y técnicas para evaluar dichas soluciones. A la larga, generalmente existen muchas formas de resolver un problema determinado. Encontrar una solución y decidir si es una buena solución, son tareas que realizaremos una y otra vez.”

"Introduction to algorithms", Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

“(...) No existe una estructura de datos que funcione bien para todos los propósitos, por lo tanto es importante conocer las fortalezas y limitaciones de varias de ellas. ” ibidem

¿Cómo determinamos qué estructura de datos y qué algoritmos son mejores para solucionar un problema determinado?

No vamos a estudiar todas estructuras de datos y todos los tipos de algoritmos. El propósito de la materia es construir herramientas conceptuales y prácticas que aporten a responder estas preguntas.

ESTRUCTURAS DE DATOS Y ALGORITMOS II

Profesores:

EMANUEL BORDA

FERNANDO MARTÍNEZ

Página:

EDYA2UPE.GITHUB.IO

TEMAS Y FECHAS

Clase	Día	Tema	Practica
1	07/04/2015	Listas	#1 Listas
2	14/04/2015	Arboles Binarios	#2 Arboles Binarios
3	21/04/2015	Arboles Generales	#3 Arboles Generales
4	21/04/2015	Arboles AVL	
5	28/04/2015	Tiempo de Ejecución	#4 Tiempo de Ejecución
6	05/05/2015	Tiempo de Ejecucion	
7	12/05/2015	Grafos	#5 Grafos
8	19/05/2015	Grafos	
9	26/05/2015	Consulta Parcial/ Clase repaso	
10	02/06/2015	Parcial 1era Fecha	
11	09/06/2015	Muestra Parcial y Consulta	
12	16/06/2015	1er Recuperatorio	
13	23/06/2015	Muestra 1er Rec y consulta	
14	30/06/2015	2do Recuperatorio	

LISTAS

LISTAS

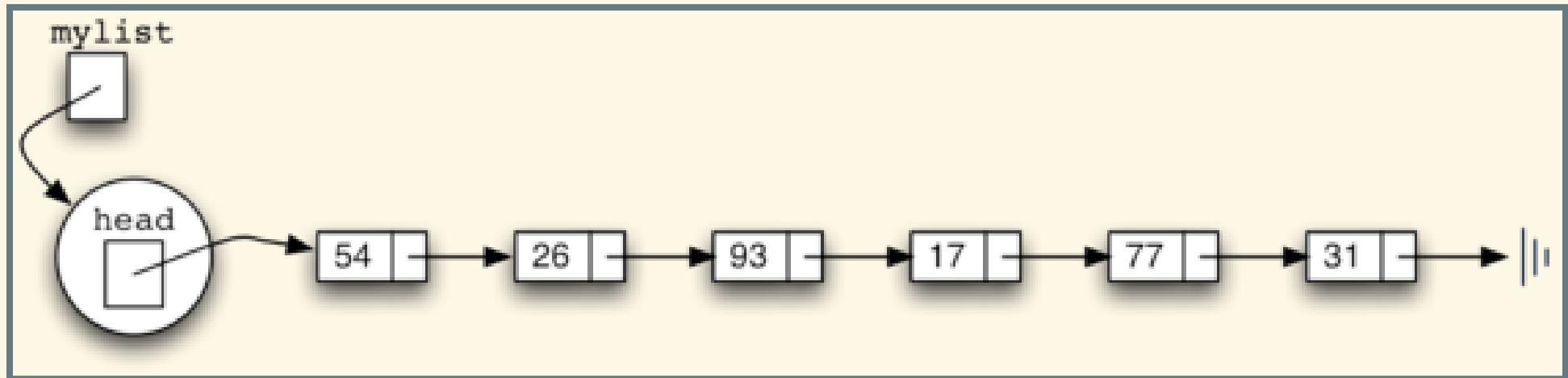
¿Qué es una lista?

DEFINICIÓN

Una lista es una colección de elementos donde cada uno mantiene una posición relativa respecto a los otros. Es decir, entre los elementos que la componen existe una relación de orden (primero, segundo, tercero... último).

Estructura lineal, tiene un principio y un final.

Es una estructura de datos simple que puede servir para construir otras estructuras de datos más complejas



¿Qué cosas podemos hacer con una lista?

¿de que manera podemos acceder a qué datos?

¿Cómo la modificamos?

OPERACIONES SOBRE LISTAS

- size
- empty
- first()
- last()
- get(index)
- find(item)
- index(item)
- includes(item)

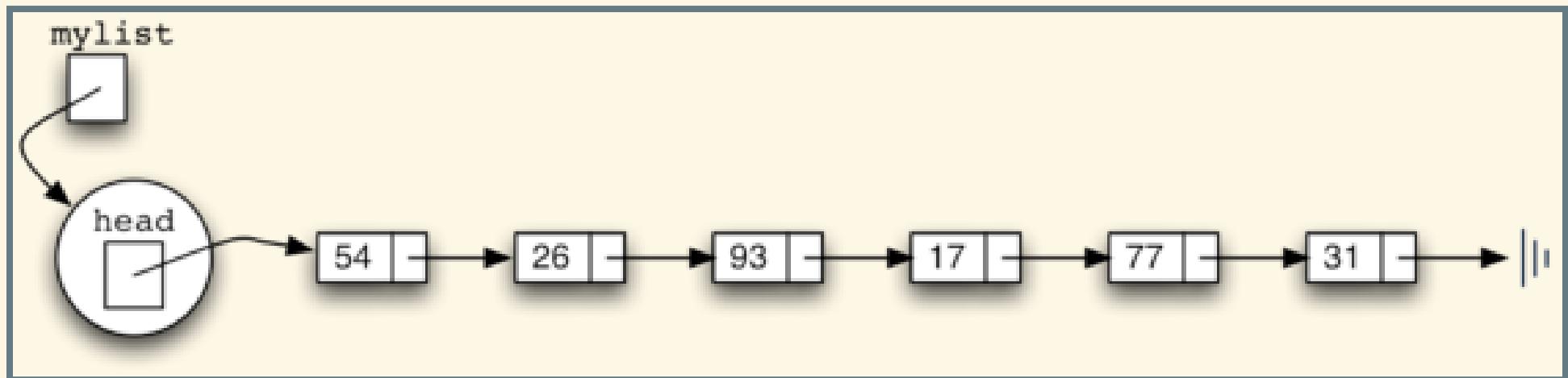
- insert(item, position)
- append(item)
- prepend(item)
- remove(item)
- clear()

TIPOS DE LISTAS

Dependiendo cómo se organicen los elementos podemos plantear distintos tipos de listas:

- lista enlazada simple (linked list)
- lista enlazada doble (double linked list)
- lista circular (circular list)
- lista no ordenada (unordered)
- lista ordenada (sorted)

LISTA ENLAZADA SIMPLE (LINKED LIST)

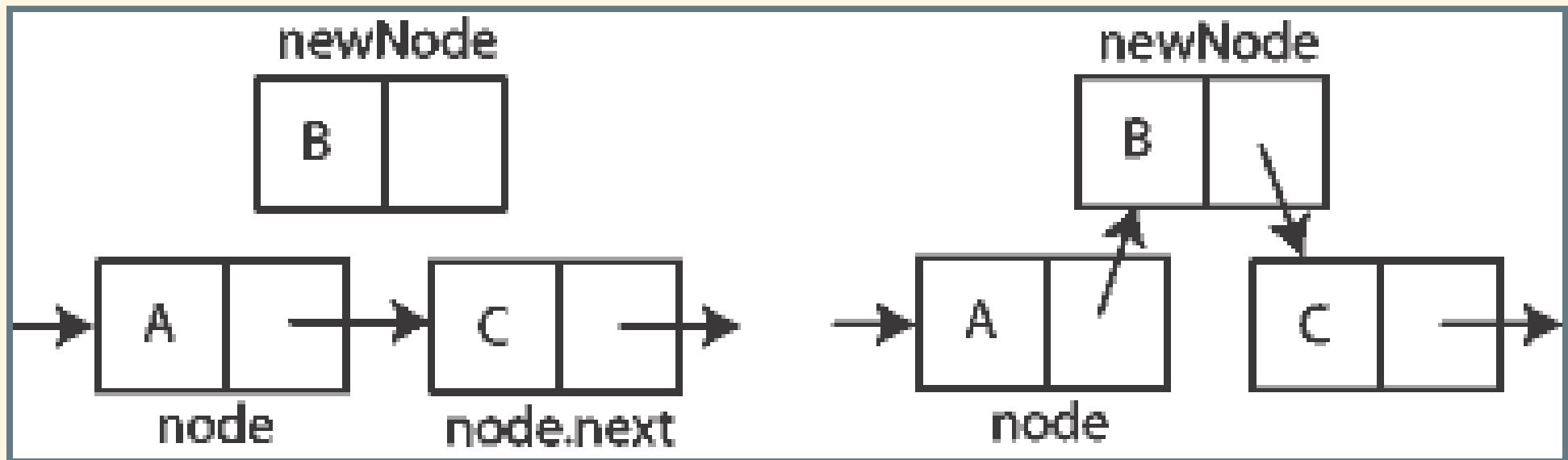


Analicemos algunas operaciones...

- recorrido/búsqueda
- inserción
- eliminación

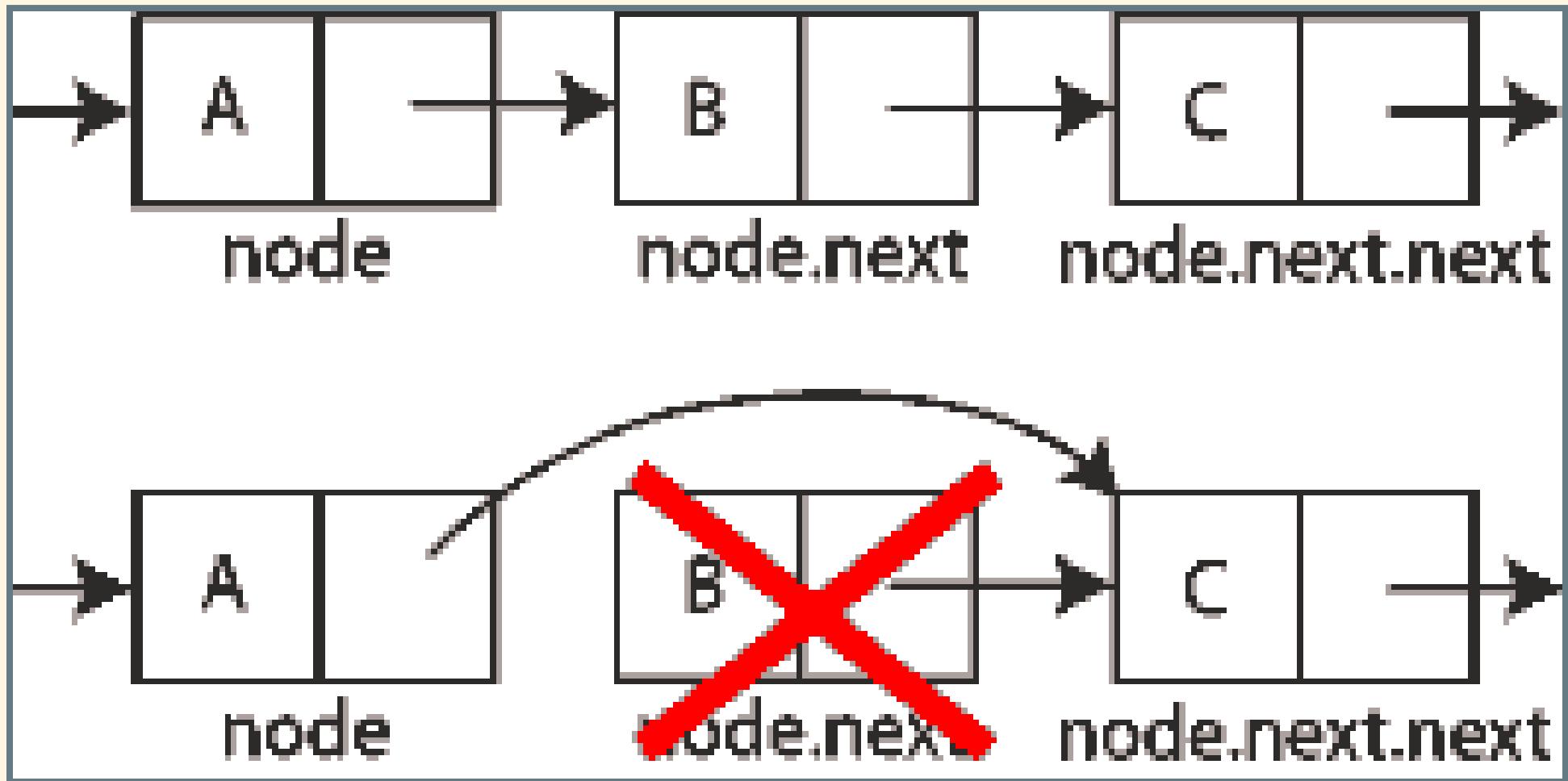
LISTA ENLAZADA SIMPLE (LINKED LIST)

INSERTAR

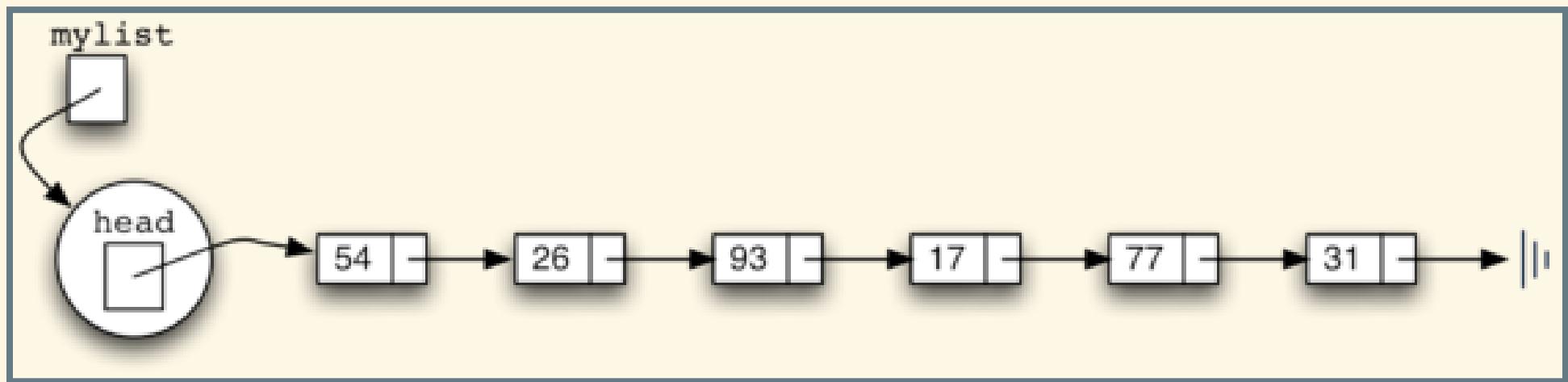


LISTA ENLAZADA SIMPLE (LINKED LIST)

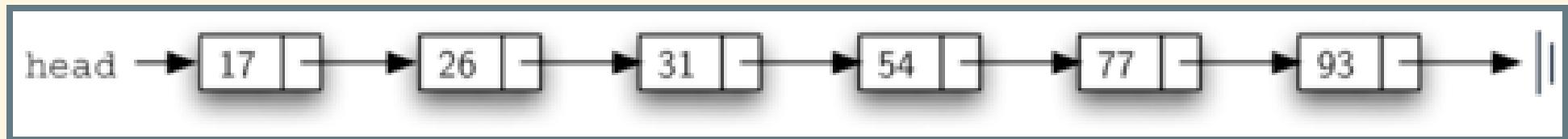
REMOVER



LISTA NO ORDENADA (UNORDERED LIST)



LISTA ORDENADA (SORTED LIST)



Comparemos algunas operaciones

¿Qué pasa con la operación `insert(item, index)`?

OTRAS ESTRUCTURAS LINEALES

- pilas (stack)
- colas (queue)
- colas dobles (dequeue)
- arreglos (array) (estáticos y dinámicos)

PILAS

operaciones:

- push(item)
- pop()
- top()

COLAS



COLAS

operaciones:

- push(item)
- pop()
- top() (o first())
- last()

podemos decir que tanto pilas como colas son un
subconjunto de las listas

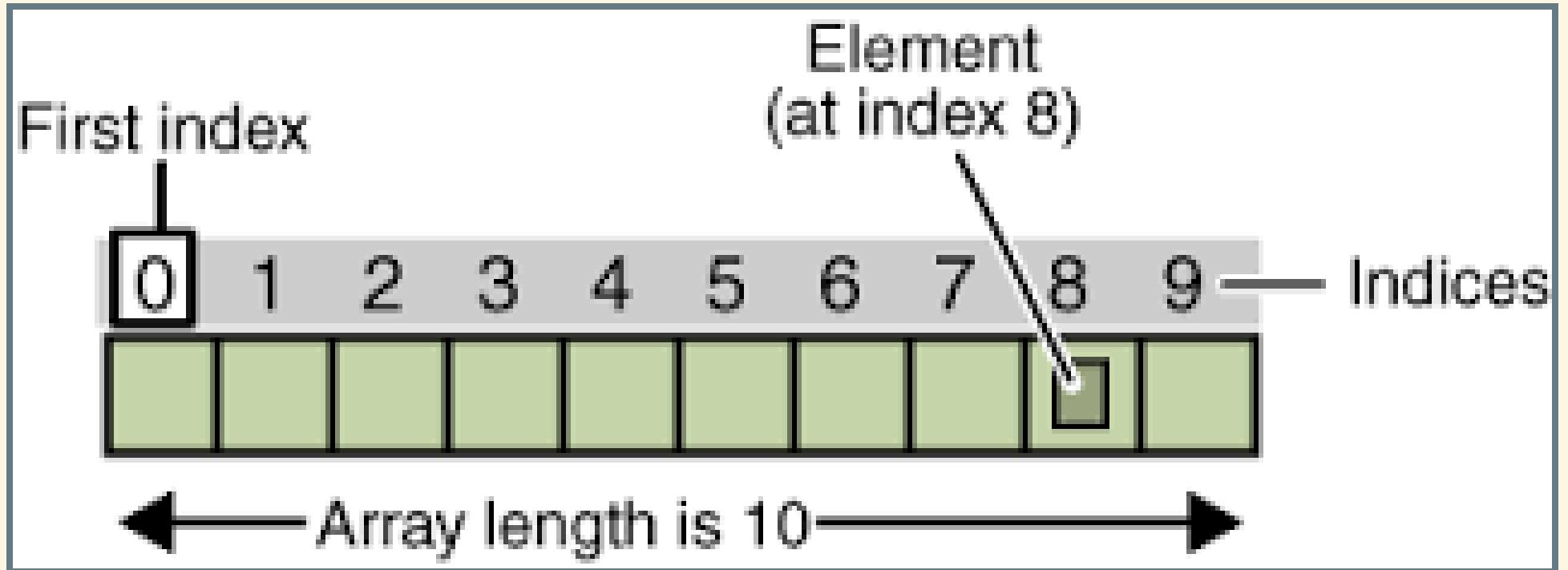
ARREGLOS (ARRAY)

Estructura de datos de bajo nivel

Almacena un conjunto de objetos en posiciones contiguas de memoria.

Indexada. Es decir puede accederse a cada uno de los elementos mediante un índice (suelen ser numéricos)

Pueden ser estáticos o dinámicos.



operaciones básicas:

- acceder a un valor
(suele escribirse $a[1]$)
- setear un valor
(suele escribirse como una asignacion $a[1] = \text{item}$)
- size o length

Entre las propiedades mas importantes encontramos: *acceso directo*

(simplemente es necesario acceder a la posicion de memoria que contiene el valor que quiero leer)

comparamos la operación de inserción en un arreglo y en una lista enlazada simple