



Instituto Politécnico Nacional



Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas

Diferencias entre LinkedList y ArrayList

Programación Orientada a Objetos

Roberto Oswaldo Cruz Leija

Aranza Miranda Montellano

Fecha de entrega:

24/10/2019

Diferencias entre LinkedList y ArrayList

La principal diferencia es un tema de implementación ArrayList implementa la lista con arreglos lineales, lo que la hace muy eficiente para recorrer pero ineficiente para insertar o eliminar elementos que se no encuentren en el final. En cambio LinkedList es una lista de elementos lineales (en realidad es una lista doblemente enlazada) lo que la hace lenta para recorrer pero sumamente eficiente para insertar o eliminar elementos. Esta es la principal diferencia y hay que recordarla siempre.

Por lo tanto ArrayList:

- Usa internamente un arreglo dinámico para almacenar los elementos.
- Proporciona una manipulación lenta
- Es la mejor opción para almacenar y acceder a datos o elementos consecutivos.

Por lo tanto LinkedList:

- Proporciona una manipulación más rápida porque utiliza una lista doblemente enlazada.
- Se puede utilizar como lista y cola porque implementa interfaz de List, Deque y Queue.
- Es mejor para manipulación de elementos, es decir, para insertar y eliminar elementos.

Por lo tanto si necesitamos agregar elementos consecutivos y recorrerlos debemos utilizar ArrayList y si en cambio necesitamos insertar y eliminar valores no consecutivos debemos usar LinkedList.

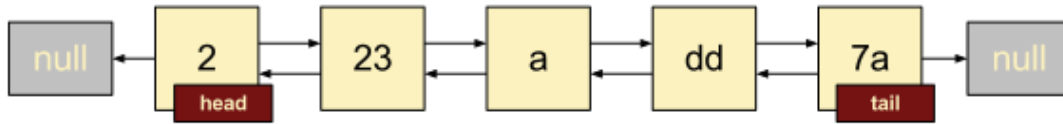
LinkedList permite eliminar e insertar elementos en tiempo constante usando iteradores, pero el acceso es secuencial por lo que encontrar un elemento toma un tiempo proporcional al tamaño de la lista.

Normalmente la complejidad de esa operación promedio sería $O(n/2)$ sin embargo usar una lista doblemente ligada el recorrido puede ocurrir desde el principio o el final de la lista por lo tanto resulta en $O(n/4)$.

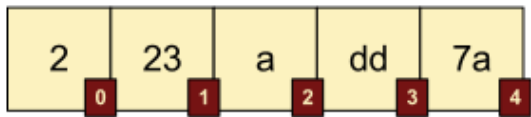
Por otro lado ArrayList ofrece acceso en tiempo constante $O(1)$, pero si quieres añadir o remover un elemento en cualquier posición que no sea la última es necesario mover elementos. Además si el arreglo ya está lleno es necesario crear uno nuevo con mayor capacidad y copiar los elementos existentes.

Array vs. Linked List

Linked List



Array



Métodos LinkedList:

addAll(): Agrega una lista.

addLast(): Agrega un nuevo nodo.

clear(): Elimina todos los elementos de una lista.

size(): Retorna un entero (El cual indica el número de elementos de una lista).

Métodos ArrayList:

add(): Añade un elemento en el final.

remove(): Borra un elemento de la lista.

clear(): Elimina todos los elementos de una lista.

iterator(): Devuelve un iterador para recorrer la lista.

get(): Devuelve el elemento en el índice indicado.

NIVEL DE COMPLEJIDAD:

ArrayList

Operación	Complejidad Promedio
get	O(1)
add(E element)	O(1)
add(int index, E element)	O(n/2)
remove(int index)	O(n/2)
Iterator.remove()	O(n/2)
remove(int index)	O(n/2)

LinkedList

Operación	Complejidad Promedio
get	O(n/4)
add(E element)	O(1)
add(int index, E element)	O(n/4)
remove(int index)	O(n/4)
Iterator.remove()	O(1)
ListIterator.add(E element)	O(1)

LinkedList:

Ventajas:

- Añadir y remover elementos con un iterador.
- Añadir y remover elementos al final de la lista.

Desventajas:

- Uso de memoria adicional por las referencias a los elementos anterior y siguiente.
- El acceso a los elementos depende del tamaño de la lista.

ArrayList:

Ventajas:

- Añadir elementos.
- Acceso a elementos.

Desventajas:

- Costos adicionales al añadir o remover elementos.
- La cantidad de memoria considera la capacidad definida para el ArrayList, aunque no contenga elementos.

Conclusión:

En la práctica la mayoría de las ocasiones es mejor usar ArrayList porque el tiempo de las operaciones y uso de memoria es menor que en LinkedList, de manera simple: si no sabes cual usar usa ArrayList.

Bibliografía:

[2018] Goette Emanuel. "Diferencia entre ArrayList y LinkedList en Java". Recuperado de: <https://emanuelpeg.blogspot.com/2018/01/diferencia-entre-arraylist-y-linkedlist.html>

[2014]. "Métodos y funciones de ArrayList Java". *Disco duro roer*. Recuperado de: <https://www.discoduroderoer.es/metodos-y-funciones-arraylist-de-java/>

[2014]. "Java: ArrayList y LinkedList". *Programación gratis*. Recuperado de: <https://programaciongratis.wordpress.com/2014/08/18/java-arraylist-y-linkedlist/>

[2016]. "Diferencia entre ArrayList y LinkedList". Recuperado de: <http://www.enrique7mc.com/2016/07/diferencia-entre-arraylist-y-linkedlist/>