Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas

Investigación:

Diferencias entre ArrayList y LinkedList

Ingeniería en Sistemas Computacionales.

Programación Orientada a Objetos

Profesor: Roberto Oswaldo Cruz Leija

Elaborada por:

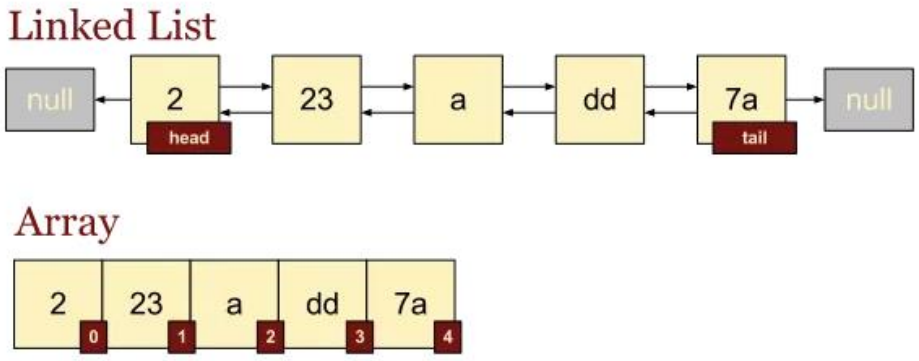
Rogelio Evenicer Valle Robles

2CM1

24/10/2019

ArrayList y LinkedList

LinkedList y ArrayList son dos diferentes implementaciones de la interfaz list. LinkedList usa internamente una lista doblemente ligada, mientras que ArrayList usa un arreglo que cambia de tamaño dinámicamente.



LinkedList permite eliminar e insertar elementos en tiempos 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.

LinkedList permite inserciones o eliminaciones de tiempo constante utilizando iteradores, pero solo acceso secuencial de elementos. En otras palabras, puede recorrer la lista hacia adelante o hacia atrás, pero encontrar un puesto en la lista lleva tiempo proporcional al tamaño de la lista. Javadoc dice que "las operaciones que indexan en la lista atravesarán la lista desde el principio o el final, lo que esté más cerca", por lo que esos métodos son O (n) (n/4 pasos) en promedio, aunque O (1) para index = 0.

ArrayList por otro lado, permite un acceso de lectura aleatorio rápido, para que puedas agarrar cualquier elemento en tiempo constante. Pero agregar o eliminar desde cualquier lugar menos el final requiere desplazar todos los últimos elementos, ya sea para abrir o llenar el espacio. Además, si agrega más elementos que la capacidad de la matriz subyacente, se asigna una nueva matriz (1.5 veces el tamaño), y la matriz anterior se copia a la nueva, por lo que agregar a un ArrayList es O (n) en el peor caso, pero constante en promedio.

Complejidad

-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)**

-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)

Estos métodos son aplicables a las clases genéricas de Java LinkedList y ArrayList sin embargo estos tipos no son exclusivos de Java, es posible encontrar implementaciones en otros lenguajes de programación, por ejemplo, en C# existen LinkedList y List.

Ventajas y desventajas

-LinkedList

Ventajas	Desventajas
Añadir y remover elementos con un iterador	Uso de memoria adicional por las referencias a los elementos anterior y siguiente
Añadir y remover elementos al final de la lista	El acceso a los elementos depende del tamaño de la lista

-ArrayList

Ventajas	Desventajas
Añadir elementos	Costos adicionales al añadir o remover elementos
Acceso a elementos	La cantidad de memoria considera la capacidad definida para el ArrayList, aunque no contenga elementos

¿Cuál debo usar?

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.

Eso no quiere decir que LinkedList nunca se utilice, existen algunos casos muy específicos donde es la mejor opción, por ejemplo, la pila de llamadas del lenguaje C está implementada usando una estructura de datos con estas características.

Referencias

http://www.enrique7mc.com/2016/07/diferencia-entre-arraylist-y-linkedlist/

https://es.stackoverflow.com/questions/172954/cuando-es-mejor-usar-linkedlist-y-cuando-arraylist