

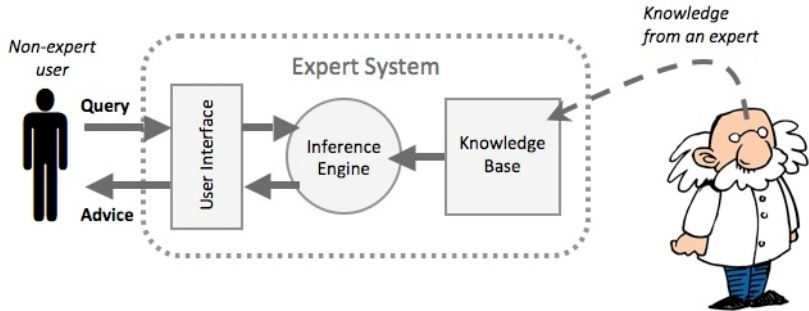
Data Structures I : Implementation of lists



Disclaimer: Keep alcohol out of the hands of minors.

- 40 ml Vodka
- 15 ml Cointreau
- 15 ml Lime juice
- 30 ml Cranberry juice





<https://www.youtube.com/watch?v=uWEahgy3Iyc>

- **Insertion** in an array is slow ($O(n)$); insertion in a linked list is fast ($O(1)$).
- Random access in an array is fast ($O(1)$); random access in a linked list is slow ($O(n)$).
- Backward traversal of Singly-linked lists is slow ($O(n^2)$); for Doubly-linked lists and arrays is fast ($O(n)$).

- **Insertion** in an array is slow ($O(n)$); insertion in a linked list is fast ($O(1)$).
- **Random access** in an array is fast ($O(1)$); random access in a linked list is slow ($O(n)$).
- Backward traversal of Singly-linked lists is slow ($O(n^2)$); for Doubly-linked lists and arrays is fast ($O(n)$).

- **Insertion** in an array is slow ($O(n)$); insertion in a linked list is fast ($O(1)$).
- **Random access** in an array is fast ($O(1)$); random access in a linked list is slow ($O(n)$).
- **Backward traversal** of Singly-linked lists is slow ($O(n^2)$); for Doubly-linked lists and arrays is fast ($O(n)$).

Binky pointer fun!

`http://www.cs.stanford.edu/cslibrary/
PointerFunJavaBig.avi`




```
class Link {  
    public int data; //data  
    public Link next; //ref. to next link  
    public Link(int data); //constructor  
}
```



```
class LinkedList
{
private Link first;
public void LinkedList(); // constructor
public void insertFirst(int data);
...
}
```

Taken from [Laf98].

- 1 `int size()`
- 2 `void insertFirst(int data)`
- 3 `void deleteFirst()`
- 4 `void deleteLast()`
- 5 `void insertLast(int data)`
- 6 `boolean contains(int data)`
- 7 `int get(int index)...`

Taken from [Laf98].

- 1 `int size()`
- 2 `void insertFirst(int data)`
- 3 `void deleteFirst()`
- 4 `void deleteLast()`
- 5 `void insertLast(int data)`
- 6 `boolean contains(int data)`
- 7 `int get(int index)...`

Taken from [Laf98].

- 1 `int size()`
- 2 `void insertFirst(int data)`
- 3 `void deleteFirst()`
- 4 `void deleteLast()`
- 5 `void insertLast(int data)`
- 6 `boolean contains(int data)`
- 7 `int get(int index)...`

Taken from [Laf98].

- 1 `int size()`
- 2 `void insertFirst(int data)`
- 3 `void deleteFirst()`
- 4 `void deleteLast()`
- 5 `void insertLast(int data)`
- 6 `boolean contains(int data)`
- 7 `int get(int index)...`

Taken from [Laf98].

- 1 int size()
- 2 void insertFirst(int data)
- 3 void deleteFirst()
- 4 void deleteLast()
- 5 void insertLast(int data)
- 6 boolean contains(int data)
- 7 int get(int index)...

Taken from [Laf98].

- 1 `int size()`
- 2 `void insertFirst(int data)`
- 3 `void deleteFirst()`
- 4 `void deleteLast()`
- 5 `void insertLast(int data)`
- 6 `boolean contains(int data)`
- 7 `int get(int index)...`

Taken from [Laf98].

- 1 `int size()`
- 2 `void insertFirst(int data)`
- 3 `void deleteFirst()`
- 4 `void deleteLast()`
- 5 `void insertLast(int data)`
- 6 `boolean contains(int data)`
- 7 `int get(int index)...`

Taken from [Laf98].

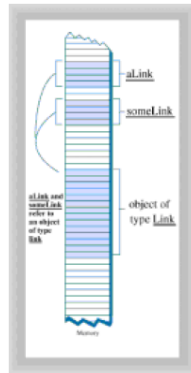


Figure: Links and references in memory. Taken from [Laf98].



Figure: Inserting a new link. Taken from [Laf98].

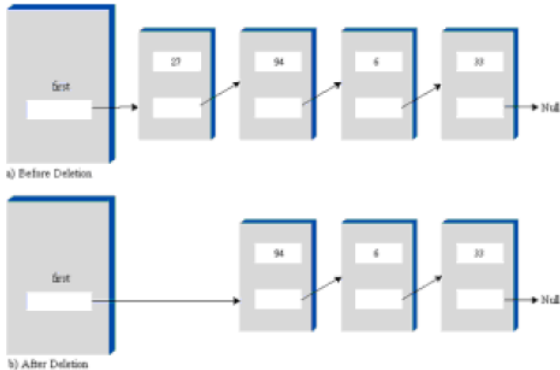


Figure: Deleting a link. Taken from [Laf98].



<http://visualgo.net/list.html>

- 1 int size()
- 2 void insertFirst(int data)
- 3 void deleteFirst()
- 4 void deleteLast()
- 5 void insertLast(int data)
- 6 boolean contains(int data)
- 7 int get(int index)...

Taken from [Laf98].

```
class ArrayList
{
    private int index;
    private int[] data;
    public void ArrayList(); // constructor
    public void insertFirst(int data);
    ...
}
```

- 1 `int size()`
- 2 `void insertFirst(int data)`
- 3 `void deleteFirst()`
- 4 `void deleteLast()`
- 5 `void insertLast(int data)`
- 6 `boolean contains(int data)`
- 7 `int get(int index)...`

Taken from [Laf98].

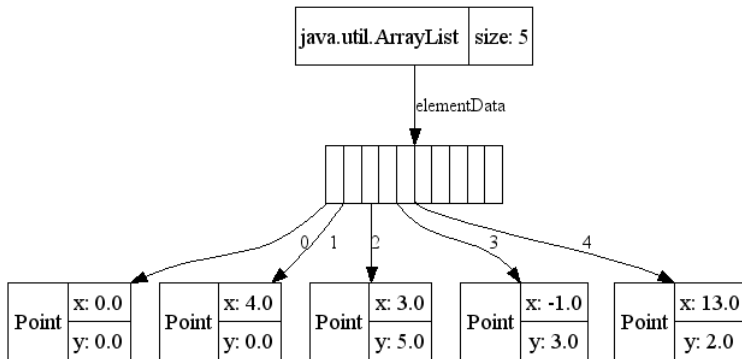


Figure: Representation of an ArrayList. Taken from <http://2.bp.blogspot.com/>

Array List Data Structure

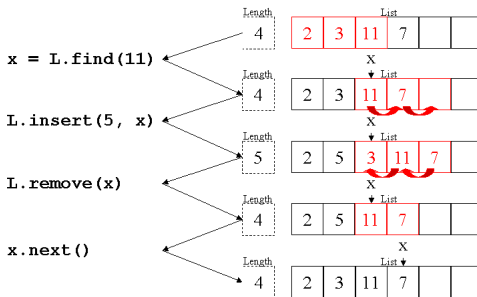


Figure: ArrayList's operations. Taken from <http://courses.cs.washington.edu/>.



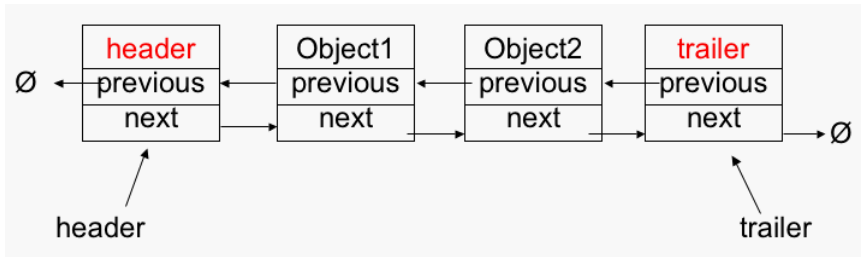
`http://www.cs.armstrong.edu/liang/animation/web/ArrayList.html`

- 1 `int size()`
- 2 `void insertFirst(int data)`
- 3 `void deleteFirst()`
- 4 `void deleteLast()`
- 5 `void insertLast(int data)`
- 6 `boolean contains(int data)`
- 7 `int get(int index)...`

Taken from [Laf98].

	insertFirst	insert(i)	deleteFirst	delete(i)	get(i)
ArrayList	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$
LinkedList	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$

Table: Complexity analysis of the operations of LinkedList and ArrayList.



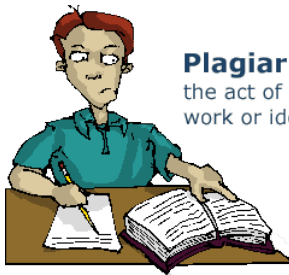
For more details, please check

- Robert Lafore. Data Structures and Algorithms in Java. Chapter 5.

- Methods for both Linked List and Array List are the same, but implementations are different.
- Linked List is more efficient for insertion and ArrayList is more efficient for random access.

- Methods for both Linked List and Array List are the same, but implementations are different.
- Linked List is more efficient for insertion and ArrayList is more efficient for random access.

- Please learn how to reference images, trademarks, videos and fragments of code.
- Avoid plagiarism



Plagiarism:

the act of presenting another's work or ideas as your own.

Figure: Figure about plagiarism, University of Malta [Uni09]



Robert Lafore.

Data Structures and Algorithms in Java.

QUE; 1 edition (8 November 2002), 1998.



University of Malta.

Plagiarism — The act of presenting another's work or ideas as your own, 2009.

[Online; accessed 29-November-2013].

- Linked Lists
 - Robert Lafore. Data Structures and Algorithms in Java. Chapter 3.

