

Data Structures I :

Linked lists



Disclaimer: Keep alcohol out of the hands of minors.

- 30 ml Cognac
- 30 ml Crème de Cacao
- 30 ml Fresh cream





https://www.youtube.com/watch?v=3ViMWbHV_cs

- 1 Number of instructions $T(n)$
- 2 Asintotic notation O
- 3 Homogeneous lineal recurrence equations



Figure: Taken from Inc. [?]



- Arrays have certain disadvantages as data storage structures
 - In an unordered array, **searching** is slow,
 - In an ordered array, **insertion** is slow.
 - In both kinds of arrays, **deletion** is slow.
 - The **size** of an array cannot be changed.



Taken from [?].

- Arrays have certain disadvantages as data storage structures
 - In an unordered array, **searching** is slow,
 - In an ordered array, **insertion** is slow.
 - In both kinds of arrays, **deletion** is slow.
 - The **size** of an array cannot be changed.



Taken from [?].

- Arrays have certain disadvantages as data storage structures
 - In an unordered array, **searching** is slow,
 - In an ordered array, **insertion** is slow.
 - In both kinds of arrays, **deletion** is slow.
 - The **size** of an array cannot be changed.



Taken from [?].

- Arrays have certain disadvantages as data storage structures
 - In an unordered array, **searching** is slow,
 - In an ordered array, **insertion** is slow.
 - In both kinds of arrays, **deletion** is slow.
 - The **size** of an array cannot be changed.



Taken from [?].

- In an **array** each item occupies a particular position.
 - This position can be directly accessed using an index number.
 - It is like a row of houses:
 - you can find a particular house using its address.
- In a **list** the only way to find a particular element is to follow along the chain of elements.
 - It is more like human relations.
 - Maybe you ask Harry where Bob is.
 - Harry does not know, but he thinks Jane might know,
 - so you go and ask Jane...

Taken from [?].

- In an **array** each item occupies a particular position.
 - This position can be directly accessed using an index number.
 - It is like a row of houses:
 - you can find a particular house using its address.
- In a **list** the only way to find a particular element is to follow along the chain of elements.
 - It is more like human relations.
 - Maybe you ask Harry where Bob is.
 - Harry does not know, but he thinks Jane might know,
 - so you go and ask Jane...

Taken from [?].

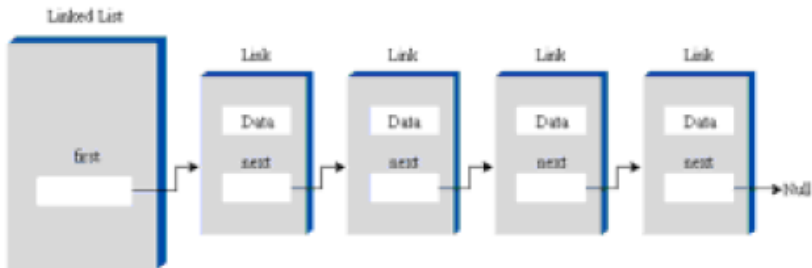


Figure: Links in a list. Taken from [?].

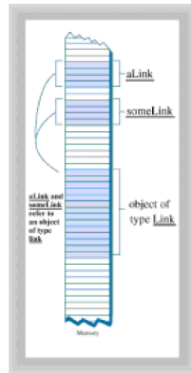


Figure: Links and references in memory. Taken from [?].



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

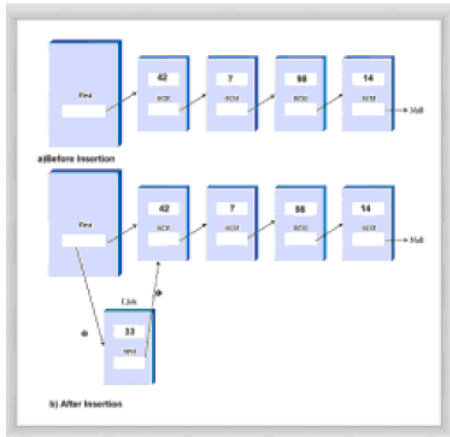


Figure: Inserting a new link. Taken from [?].

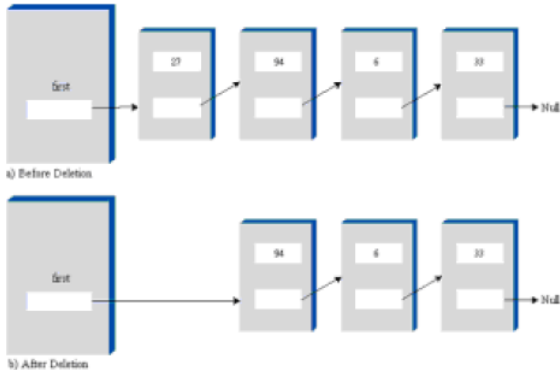


Figure: Deleting a link. Taken from [?].

- With ordinary **linked lists** is that it is difficult to traverse backward along the list.
- For example, imagine a **text editor** in which a linked list is used to store the text.
 - Each text line on the screen is stored as a String object embedded in a **link**.
 - When the editor's user moves the cursor downward on the screen, the program steps to the next **link** to manipulate or display the new line.
 - But what happens if the user moves the cursor upward?

Taken from [?].

- With ordinary **linked lists** is that it is difficult to traverse backward along the list.
- For example, imagine a **text editor** in which a linked list is used to store the text.
 - Each text line on the screen is stored as a String object embedded in a **link**.
 - When the editor's user moves the cursor downward on the screen, the program steps to the next **link** to manipulate or display the new line.
 - But what happens if the user moves the cursor upward?

Taken from [?].

- With ordinary **linked lists** is that it is difficult to traverse backward along the list.
- For example, imagine a **text editor** in which a linked list is used to store the text.
 - Each text line on the screen is stored as a String object embedded in a **link**.
 - When the editor's user moves the cursor downward on the screen, the program steps to the next **link** to manipulate or display the new line.
 - **But what happens if the user moves the cursor upward?**

Taken from [?].



Figure: A doubly-linked list. Taken from [?].

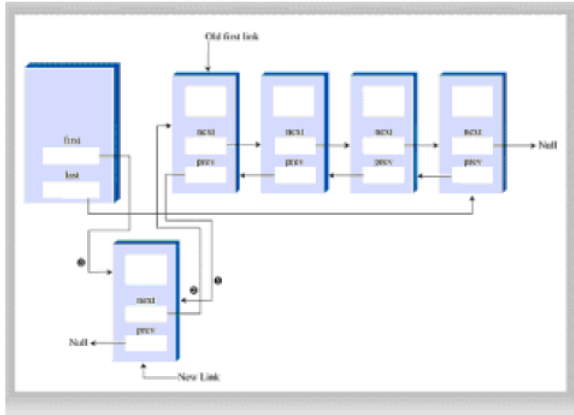


Figure: Insertion at the beginning. Taken from [?].

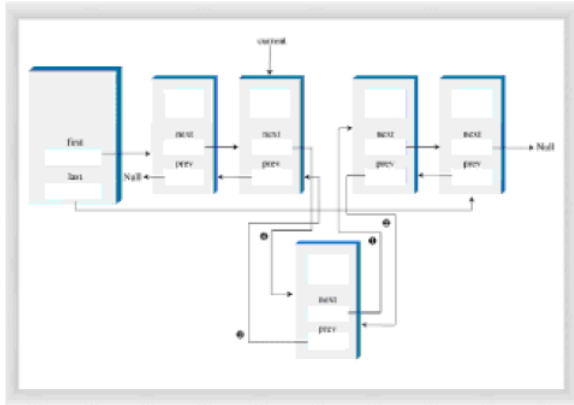


Figure: Insertion at an arbitrary location. Taken from [?].

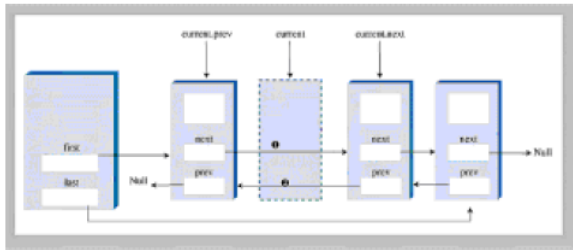


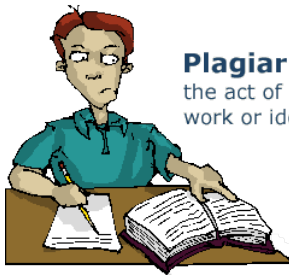
Figure: Deleting an arbitrary link. Taken from [?].

- **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 ($O(n)$).

- 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 [?]

■ Lists

- Jorge Villalobos. Introducción a las Estructuras de Datos: Aprendizaje Activo Basado en Casos. Nivel 3. Páginas 177 – 209.

