

Chapter 3

Abstract Data Type

ADT

- Set of objects together
- Set of operations

Linked List

Arrays

Stack

1st homework

Queue

Array

- fixed capacity

* Print $\rightarrow O(n)$

* find $k^{\text{th}} \rightarrow O(1)$

* insert/delete $\rightarrow O(n)$

* find k

\hookrightarrow depends

\hookrightarrow if sorted apply

binary search $O(\log n)$

\hookrightarrow if unsorted

linear

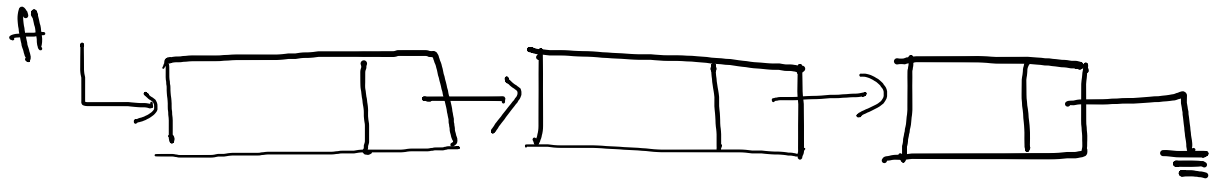
Good for

\rightarrow Add/remove happens at the end

\rightarrow Only find k^{th} is needed

\rightarrow when the application demands storing and accessing

Linked List



Each node has

Singly linked list

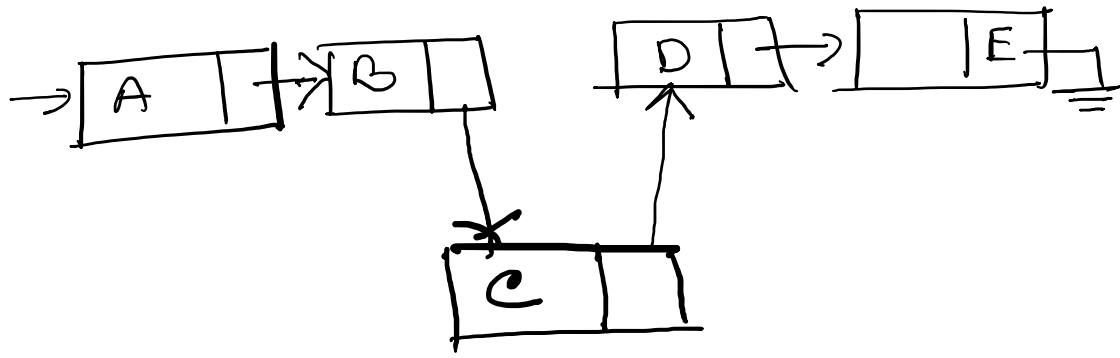
- element
- link to node containing successor → next
- next of last element is null

Operation

* find (x) → $O(n)$

* find k^{th} → $O(n)$

* insert/remove →



$O(1)$

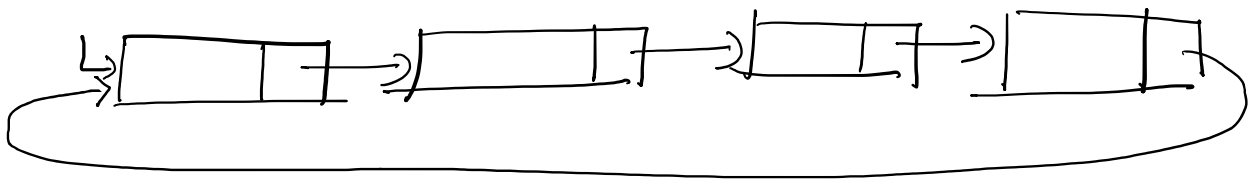
In a linked if you want to remove the second last element, what type of LL will you pick n why??

→ D. LL

→ in a D. LL you have ref to the previous node.

Array	Linked List
→ get/set take constant time	→ get/set is expensive as it starts from the beginning of the list
→ insertion/removal is expensive due to shifting behavior	→ insertion/removal operation only is cheap (the operation only)

Circular Linked List



Stack

→ Insert/Delete happens from one location only

→ top

→ end of the list

→ Push

→ Pop

→ only top element is accessible.

→ LIFO