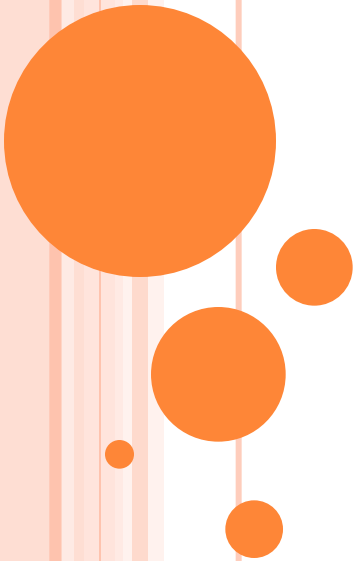
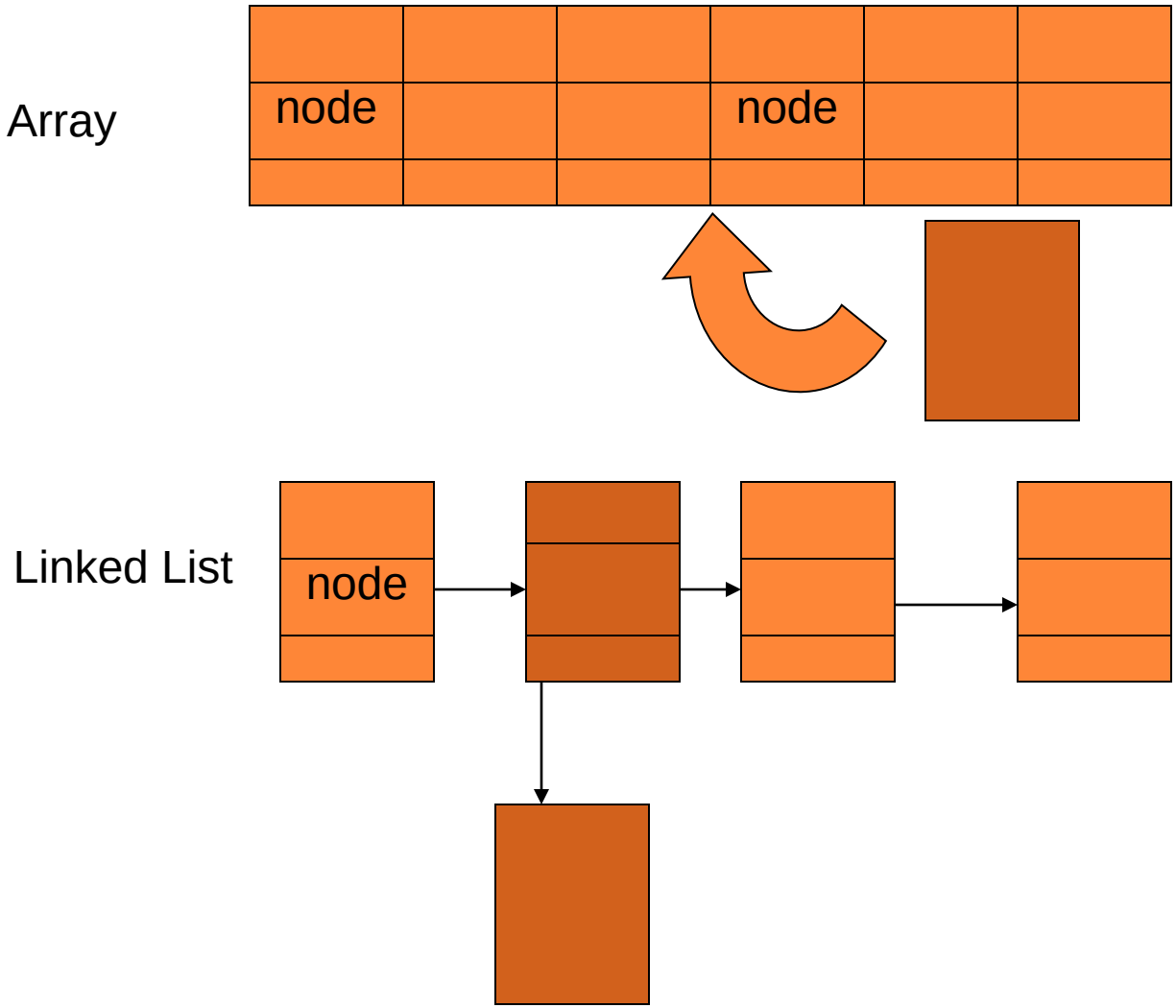


Linked Lists

Array vs Linked List



What's wrong with Array and Why lists?

- Disadvantages of arrays as storage data structures:
  - slow searching in unordered array
  - slow insertion in ordered array
  - Fixed size
- Linked lists solve some of these problems
- Linked lists are general purpose storage data structures and are versatile.

Linked Lists

- Each data item is embedded in a link.
- Each Link object contains a reference to the next link in the list of items.
- In an array items have a particular position, identified by its index.
- In a list the only way to access an item is to traverse the list
- Is LL an ADT?

Operations in a simple linked list:

- Insertion
- Deletion
- Searching or Iterating through the list to display items.

Operations in a simple linked list:

- The simplest methods are
  - insertfirst() and
  - deletefirst(),
  - where the first item in the linked list is accessed and deleted or a new item is inserted as the head or root of the list.
- To insert or delete items from any other part of the list, we need to traverse the list starting from its root and traversing till we get the item that we are looking for.

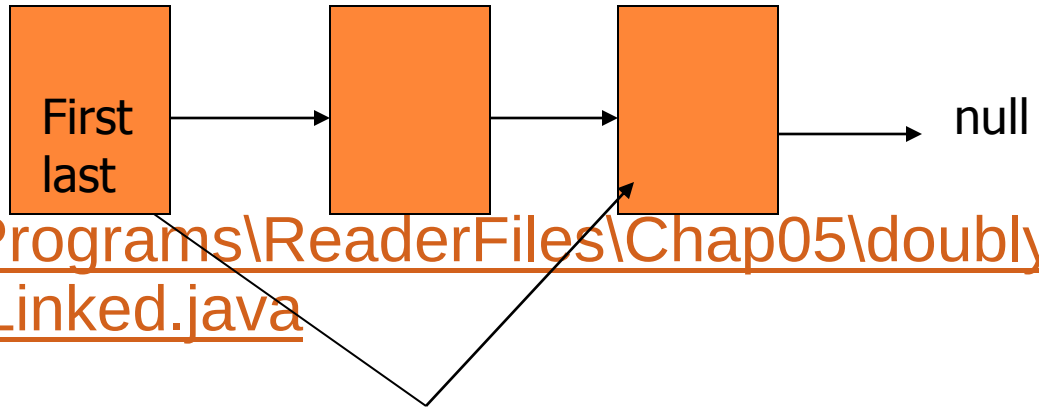
Examples

- ..\ReaderPrograms\ReaderFiles\Chap05\linkList\linkList.java
- ..\ReaderPrograms\ReaderFiles\Chap05\linkList2\linkList2.java
- Linklist2.java: delete (key), find(key)

Double-Ended Lists

- Similar to an ordinary list with the addition that a link to the last item is maintained along with that to the first.
- The reference to the last link permits to insert a new link directly at the end of the list as well as at the beginning.
- This could not be done in the ordinary linked list without traversing the whole list.
- This technique is useful in implementing the Queue where insertions are made at end and deletions from the front.

Double-Ended Lists



- ..\ReaderPrograms\ReaderFiles\Chap05\doublyLinked\doublyLinked.java

Linked List Efficiency

- Insertion and deletion at the beginning of the list are very fast, O(1).
- Finding, deleting or inserting in the list requires searching through half the items in the list on an average, requiring O(n) comparisons.
- Although arrays require same number of comparisons, the advantage lies in the fact that no items need to be moved after insertion or deletion.
- As opposed to fixed size of arrays, linked lists use exactly as much memory as is needed and can expand.

Abstract Data Types

- Focus on what the data structure does
- Ignore how it does.
- ADT is a class considered without regard to its implementation.
- Examples: Stacks and Queues
- They can also be implemented using linked lists as opposed to array in the previous chapter.

ADT Lists

- Also called linear list.
- Group of items arranged in a linear order.
- Operations supported are : insertion, deletion and read an item.
- List is defined by its interface; the specific methods used to interact with it.
- This can be implemented using arrays or linked lists.

Sorted Lists (ProrityQ)

- As the name suggests data is stored in order.
- Find and delete methods are used.
- Advantage of sorted list over sorted array is speed of insertion and its ability to expand to fill available memory.
- Efficiency:

-- Insertion and deletion of arbitrary items require O(n) comparisons.

Doubly Linked Lists

- Solves the problem of traversing backwards in an ordinary linked list.
- A link to the previous item as well as to the next item is maintained.
- The only disadvantage is that every time an item is inserted or deleted, two links have to be changed instead of one.
- A doubly-linked list can also be created as a double – ended list.
- See doublyLinked.java
- ..\ReaderPrograms\ReaderFiles\Chap05\doublyLinked\doublyLinked.java

Iterators (ADT)

- An iterator is a reference that points to a link in an associated list.
- In order to traverse a list, performing some operation on certain links it is efficient to go from one link to another, checking whether each meets the criteria and then performing the operation.
- To do this, we need a reference that can be incremented.
- This reference can be embedded in a class object.
- Objects containing references to items in data structures, used to traverse theses structures are called Iterators.

Methods in Iterator

- The iterator methods allow the user to move along the list and access the link currently pointed to.
- The following methods make the iterator more flexible:

-- reset()
-- nextLink()
-- getCurrent()
-- atEnd()
-- insertAfter()
-- insertBefore()
-- deleteCurrent()