

## Year 12 Week 1

Read the following notes and answer the questions that follow:

### Linked Lists

A 'linked list' holds data in such a way that the information in one element leads to the location of the successive element. Graphically, a linked list is represented as shown in diagram 1. This particular list contains 5 elements. Each element consists of two parts; the first contains the value required and the second contains a pointer that shows where the next element is. The pointer of the last element is a value that shows that the list ends there. We also need a pointer to show us where the list starts.

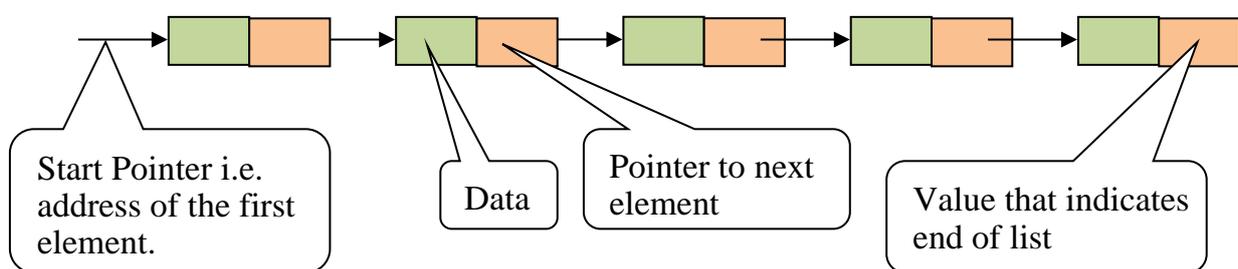


Diagram 1: A Linked List

### Example of a Linked List

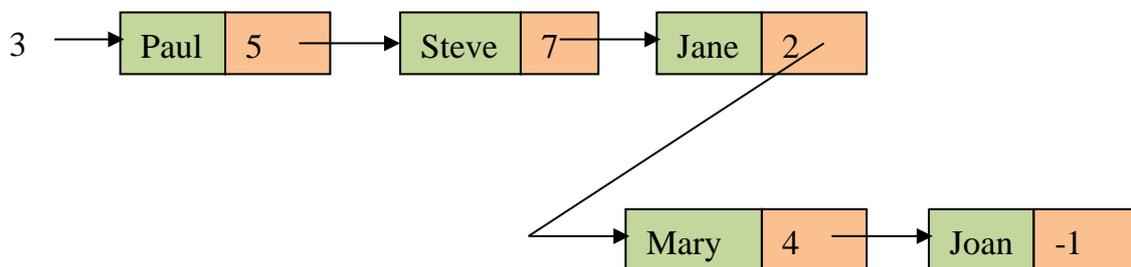


Diagram 2: A Linked List holding Names

Note that (for diagram 2):

- The address of Mary is 2.
- Paul is the first element and his address is 3.
- The end-of-list is marked -1.

### Searching in a Linked List

A linked list is a sequential data structure. Therefore when a search is made the search has to start from the first element. If the first element is not the element required we go to the second element, and so on.

## Adding an Element to a Linked List

Consider the example in diagram 2. Suppose we need to add the element 'Mark' after 'Jane'. To do this we do the following:

- Get a list element from the free list (the free list holds elements of the list that are currently not in use). (Let us assume that the free element is location 10)
- Write 'Mark' in the data section at location 10.
- Make the pointer of 'Mark' point to 'Mary' (i.e. this pointer should be 2).
- Change the pointer of 'Jane' to point to 'Mark' (the pointer part of element holding 'Jane' is changed to 10).

These steps create the linked list in diagram 3.

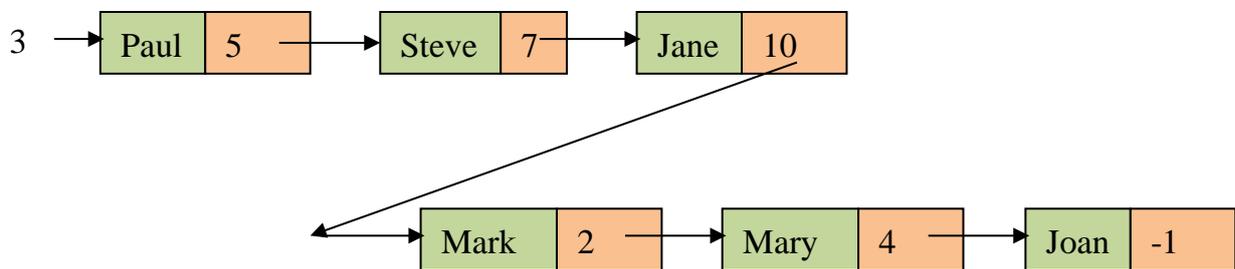


Diagram 3: Adding an Element to a List

## Deleting an Element from a Linked List

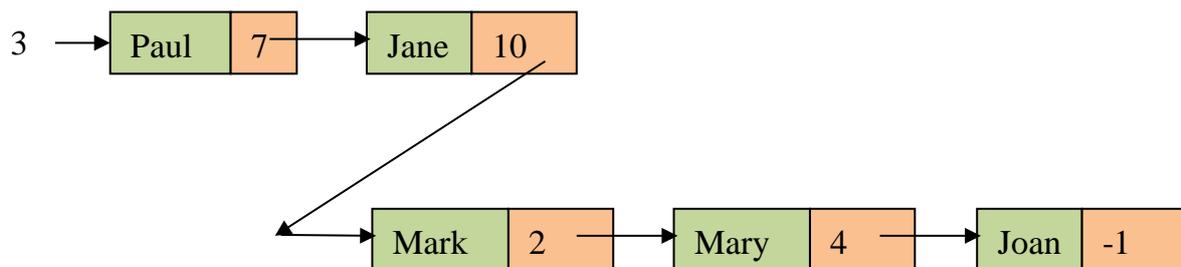


Diagram 4: Removing an Element from a List

Now 'Steve' has been removed. The following steps were taken:

- The pointer of the element holding 'Paul' now points to 'Jane'.
- The element containing 'Steve' is returned to the free list.

## An Advantage and a Disadvantage of Linked Lists

One advantage of linear lists is that an insertion or deletion of an element causes little change in the data structure holding the linked list. However one disadvantage is that searches in linked lists are only sequential.

## Circular Lists

A circular list is simply a list that has the pointer of the last element pointing to the start of the list. A diagram is shown below.

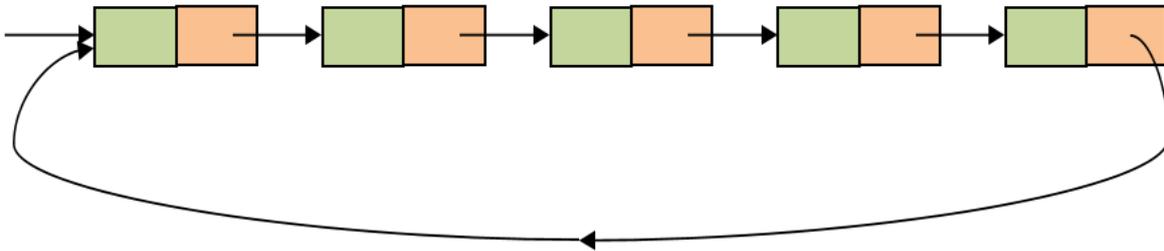


Diagram 5: A Circular List

## Double Linked Lists

A double linked list (also called two-way linked list) gives the possibility of traversing the list in two directions. In the linked lists we have seen so an element could only take you to the next element but not to the preceding element. Double linked lists can do this.

The diagram below shows how this can be done.

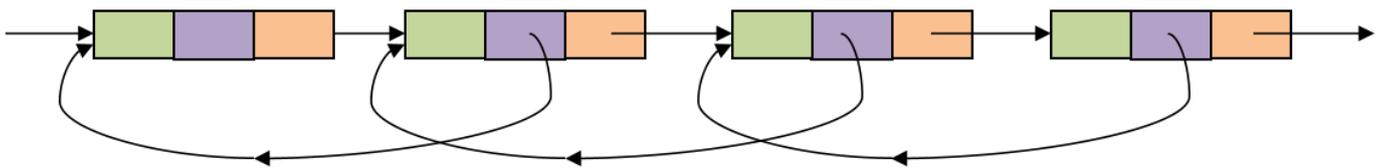
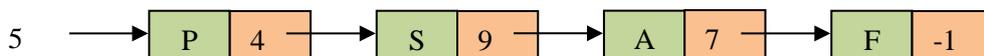


Diagram 6: A Double-Linked List

## Exercise on Linked Lists

Consider the following linked list.



- In which position (address) can you find letter A?
- In which position can you find letter P?
- A new element M is placed at location 2. This is to be inserted between S and A. Redraw the new linked list.
- Now S is removed. Redraw the linked list again.
- How would the linked list (after the modifications) appear if it was changed to a circular linked list?
- How would the same linked list appear if it was changed to a two-directional linked list?