

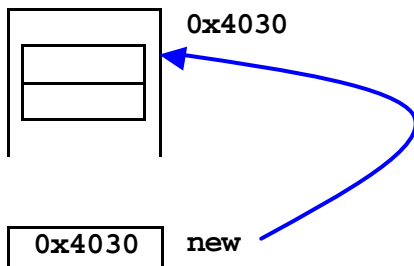
Additional Explanations – Lecture 7 - Nodes

A node is simply like an element in an array, except it is created dynamically by the program as it is running (in the heap). It is a structure that contains data, and a pointer to link it into the list/stack/queue/tree you are creating.

Create a node:

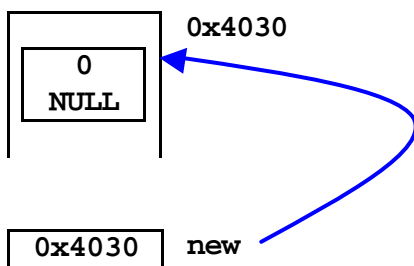
```
struct NodeRec
{
    int data;
    struct NodeRec* next;
};
typedef struct NodeRec Node;
```

```
Node* newNodePtr;
newNodePtr = (Node*)malloc(sizeof(Node));
```



Initialise a node:

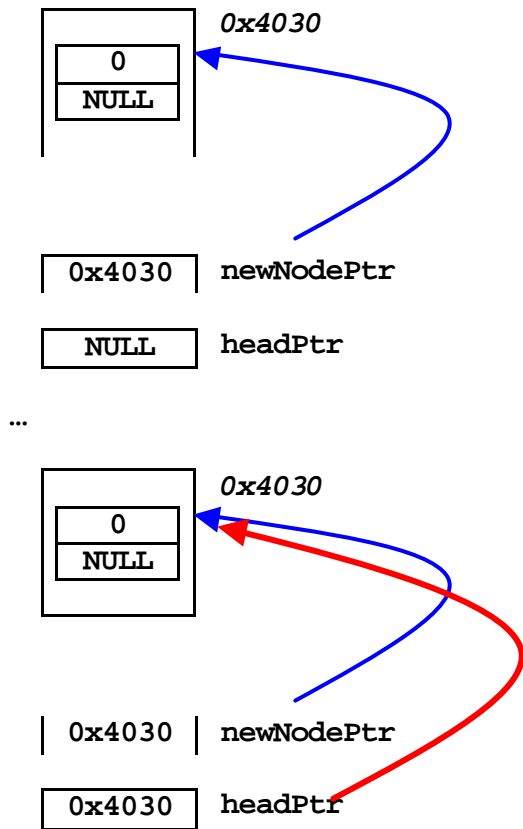
```
newNodePtr->next = NULL;
newNodePtr->data = 0;
```



CSE1303 Part A – Data Structures and Algorithms

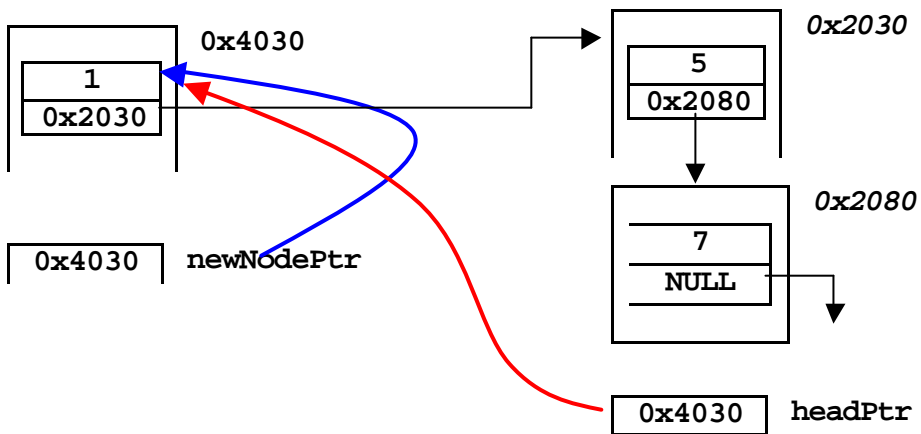
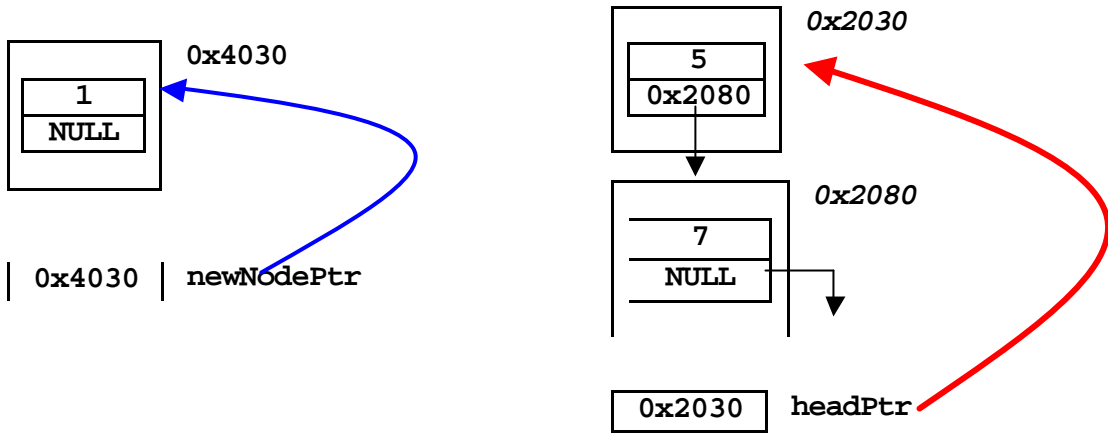
To use nodes in a list, a next pointer is used.

Empty list – insert a new node:



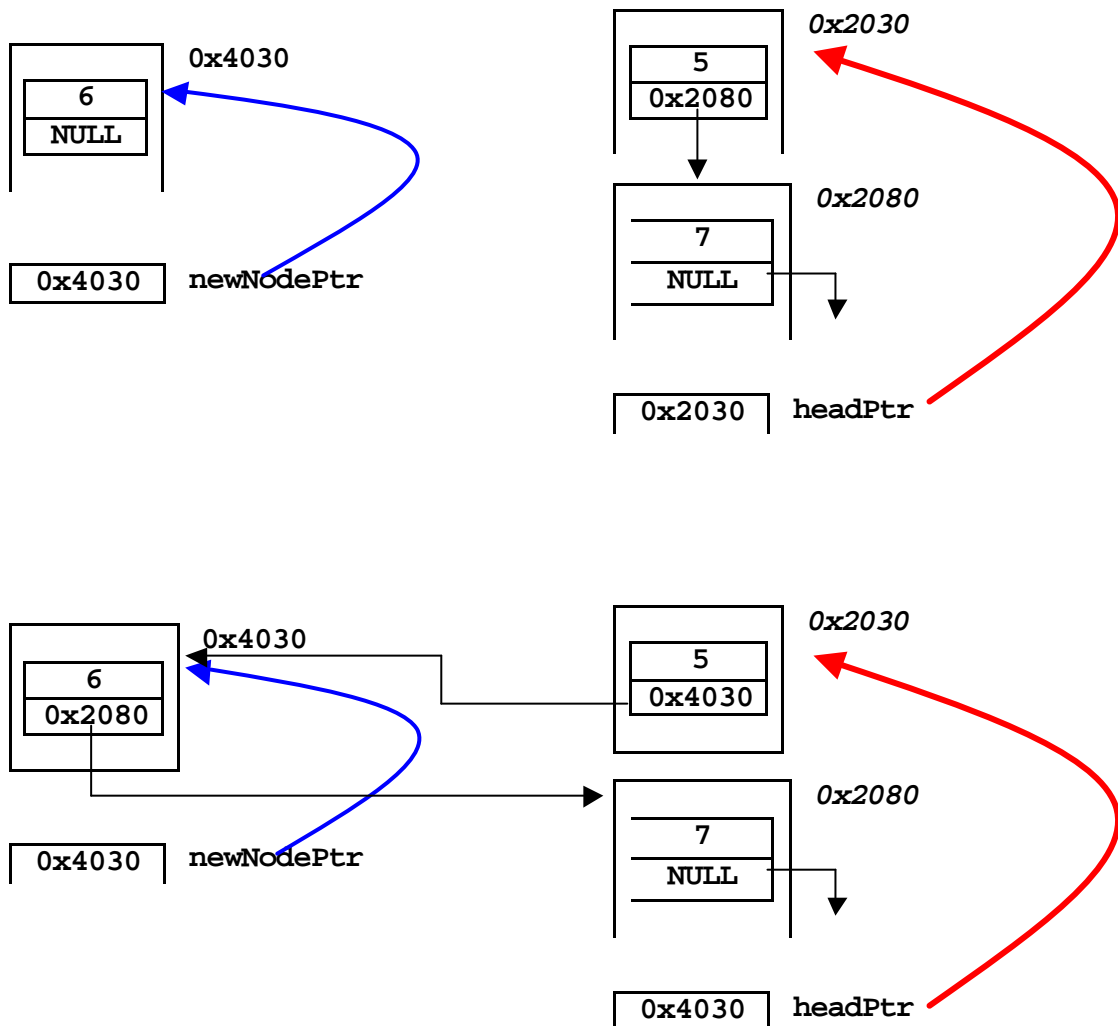
CSE1303 Part A – Data Structures and Algorithms

Insert a new node at the start of the list:



CSE1303 Part A – Data Structures and Algorithms

Insert a new node into the middle of a list:



Think about deleting from the middle of a list:

