BList Diagrams
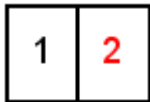
Pushing back the values $1, 2, 3, 4$, and 5 into a list with 2 items per node:
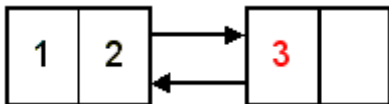
Push back **1**:

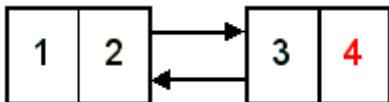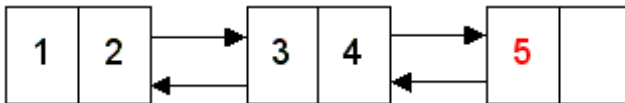Push back **2**:

Push back **3**:

Push back **4**:

Push back **5**:

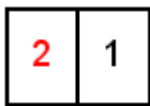This pattern will continue as you *push_back* more elements.

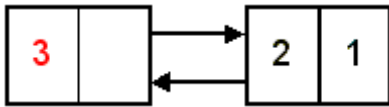Pushing front the values $1, 2, 3, 4$, and 5 into a list with 2 items per node:
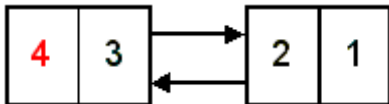
Push front **1**:
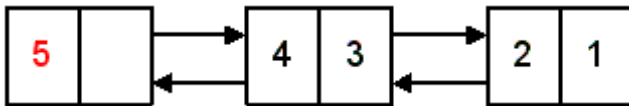

Push front **2**:

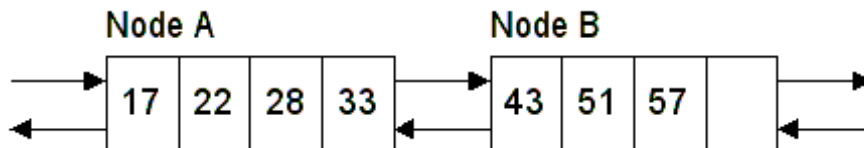**Push front 3:**

Push front 4:

Push front 5:

This pattern will continue as you *push_front* more elements.
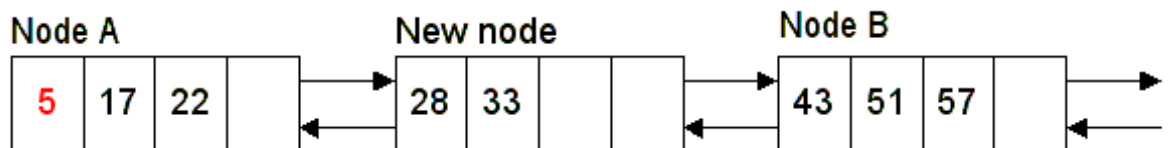
insert

Examples of inserting values into a sorted list with 4 items per node.

Inserting into a full node. Assume this is the starting point:
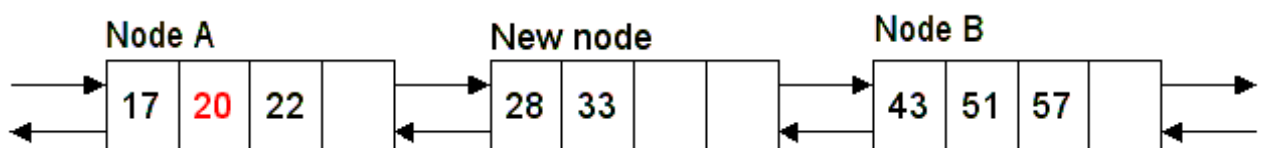

This example assumes that Node A is at the head of the list. Otherwise, it would go before Node A.
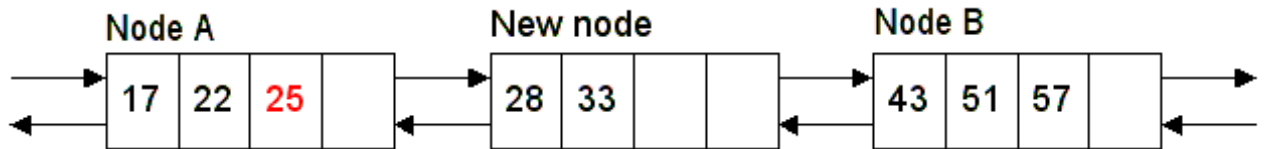
Inserting the value **5**. Since Node A is full, we split it into two nodes [17 22 __ __] and [28 33 __ __] and insert it at the front of the left node:
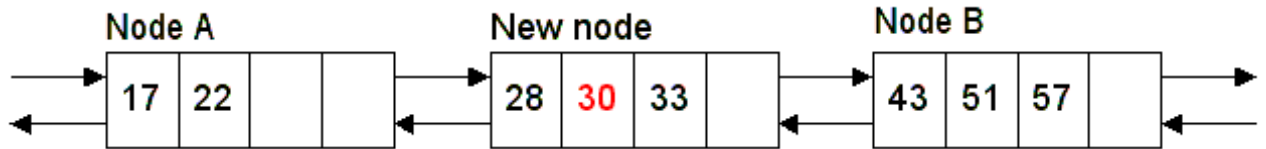

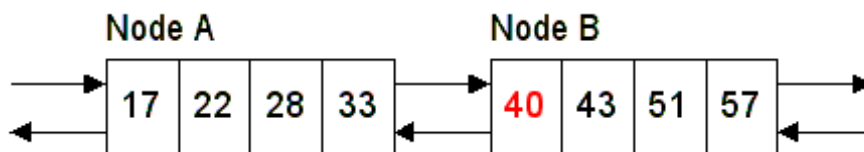Inserting the value **20** into the original list:

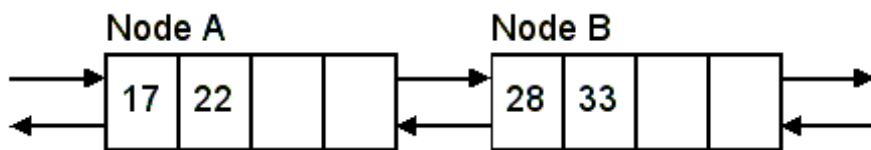Inserting the value **25** into the original list:

Inserting the value **30** into the original list:
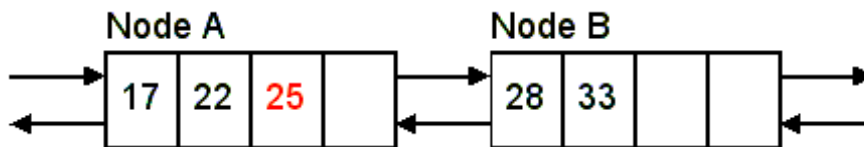

Inserting the value **40** into the original list. The reason we don't split Node A is because there is room for 40 in Node B.
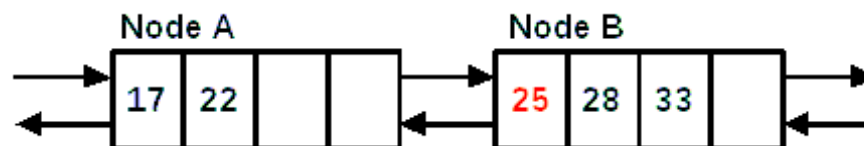

Assume we have this situation and want to insert the value **25**:

Inserting the value **25** (Correct):


Inserting the value **25** (Incorrect):

Even though the value 25 can go in either node above (both have room), we want to minimize the shifting of existing elements.
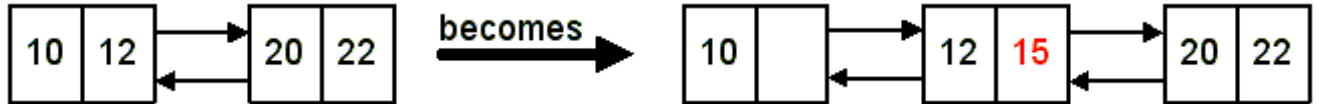
Splitting Examples

If both nodes are full and the new value is greater than the largest value in the left node and less than the smallest value in the right node, we want to **favor the left node for insertion**.

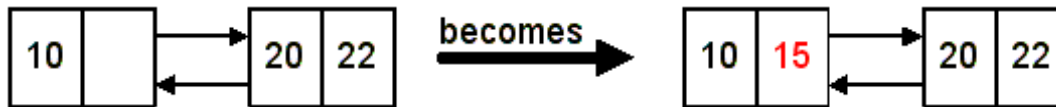Inserting the value **15** in four different cases.

1. Both the left and right nodes have room:

| 10 | | → | 20 | | **becomes** → | 10 | 15 | → | 20 | |
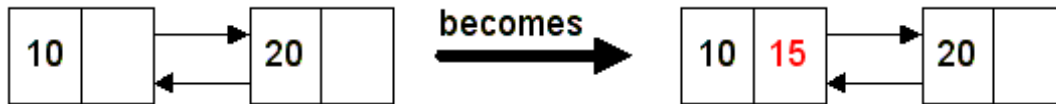
2. Both the left and right nodes are full: (Split the left node)

| 10 | 12 | → | 20 | 22 | **becomes** → | 10 | | → | 12 | 15 | → | 20 | 22 |

3. The left node has room and right node is full:

| 10 | | → | 20 | 22 | **becomes** → | 10 | 15 | → | 20 | 22 |

4. The left node is full and the right node has room:

| 10 | 12 | → | 20 | | **becomes** → | 10 | 12 | → | 15 | 20 |