

Doubly linked list

```

class node
{
public:
  
```

```

    int data;
    node * next;
    node * prev; ✓
    node (int val)
    {
  
```

```

        data = val;
        next = NULL;
        prev = NULL; ✓
    }
  
```

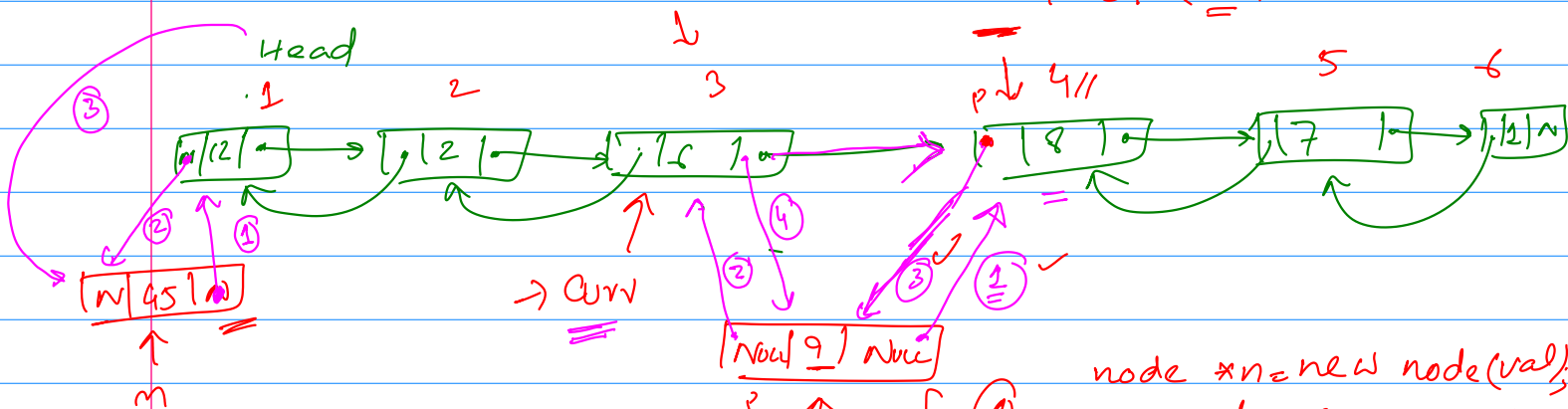
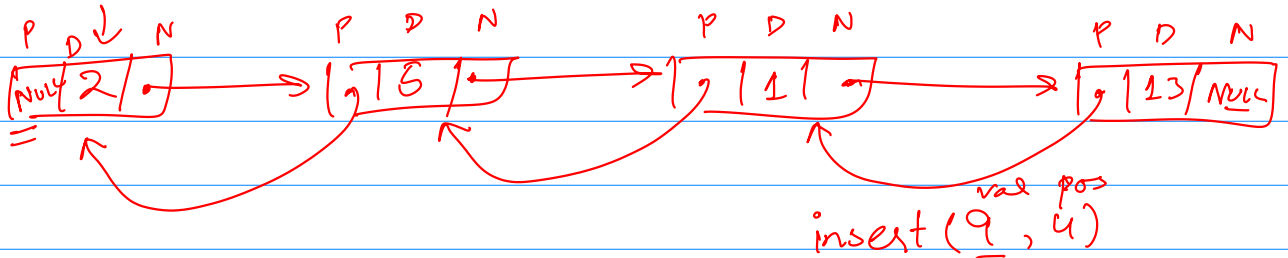
```

}; head ✓
  
```

list ✓

- Array based
- fast.
- size (fixed)
- linked list
- slow
- size (flexible)

size = 4.



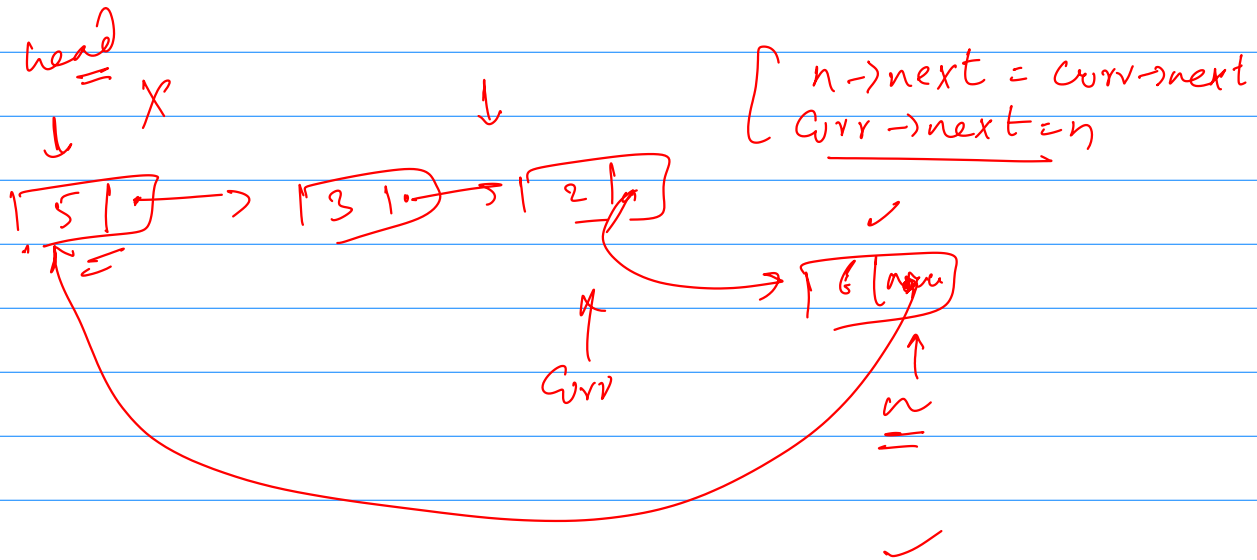
- ① $n \rightarrow next = head;$
- ② $head \rightarrow prev = n;$ (Extra)
- ③ $head = n;$

size++

```

node * n = new node(val);
n->next = curr->next;
n->prev = curr;
curr->next->prev = n;
curr->next = n;
  
```

Josephus problem circular 10 nodes



int
leader ←

Josephus (int m, int n) {

m=3

delete

return *head;

no. of nodes

[Circularly connect

