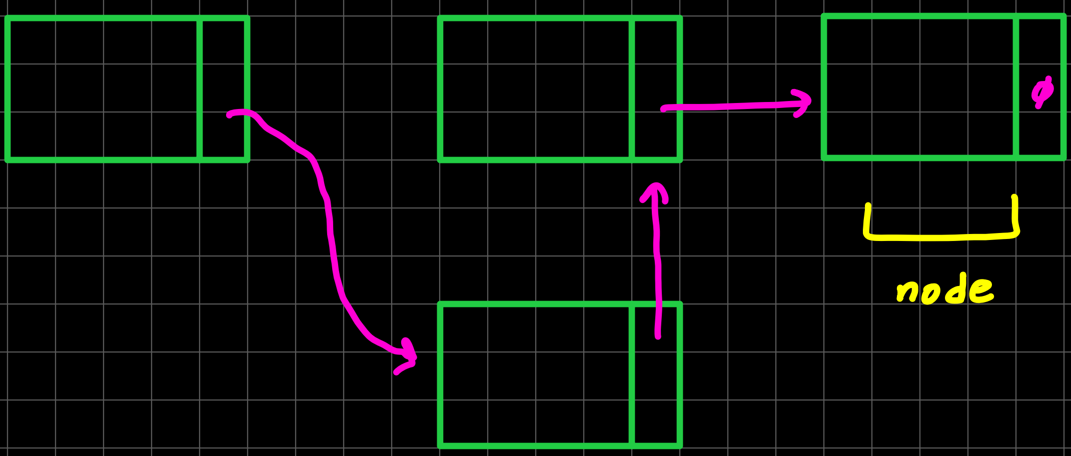
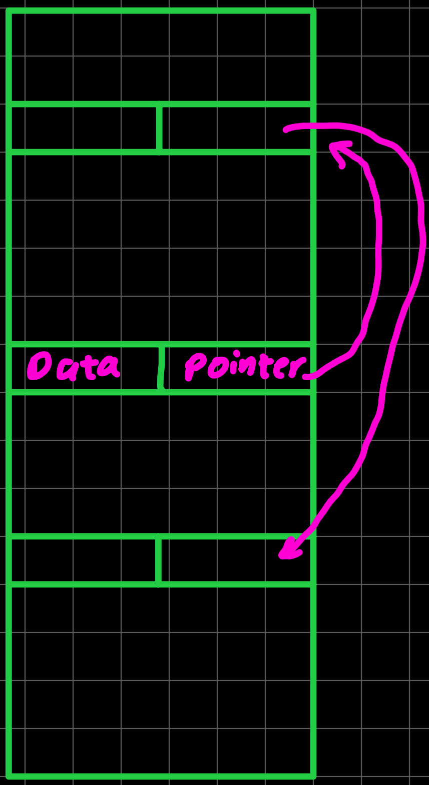
Linked List

- Segmented Memory results in an expensive option of growing an array.
  → copying into new array & then adding

- Random storage of elements in Linked List → Ideal for growing and shrinking
- whole List is traversed for searching → expensive operation

- Pointer is null if no element next.





Syntax

- Creation

```
template < class T >
class List {

    private:
            class Node {

                public:

                    T data;
                    Node pointer;

                    Node ( T _data , Node _pointer) {
```

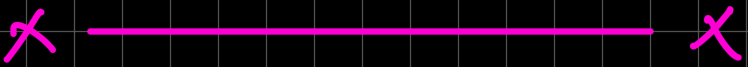
```
                data = _data;
                pointer= _pointer;
            }

            Node(){



        };

        Node* head;
    public:
        List();
```



- Insertion