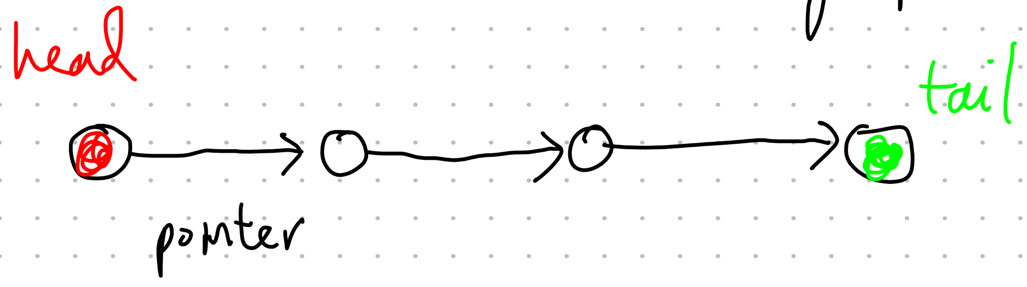
Linked Lists

Indiv. nodes that hold data
↳ linked by pointers



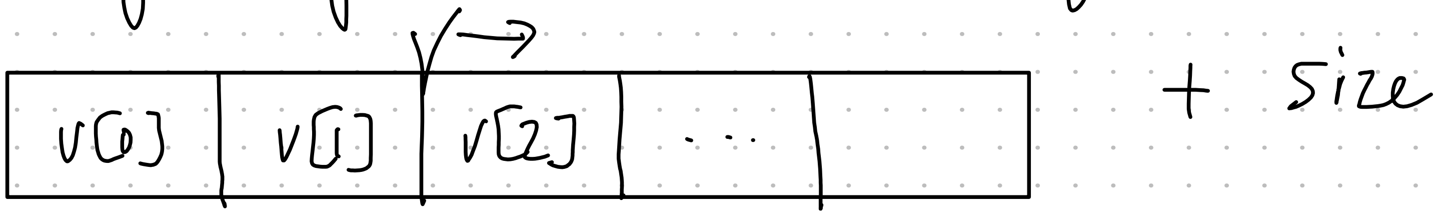
```
struct Node {
    Node * next;
    int data;
};
```

```
struct List {
    Node * head;
    Node * tail;
}
```

Vectors vs. Linked List

Vectors

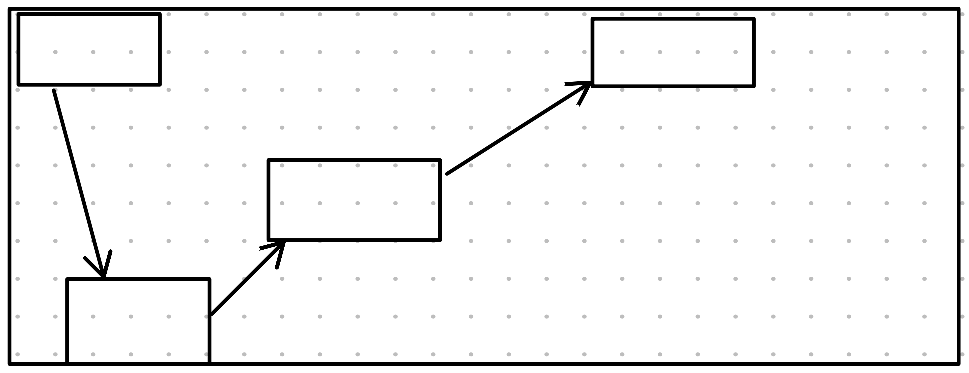
- Contiguously laid out in memory



+ Easy to access the nth element
+ No pointers means less memory overhead
- Insertion is expensive – need to shift elements →

List:

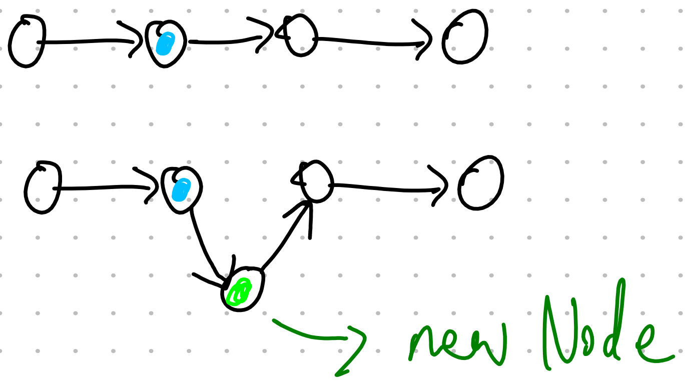
- Not contiguous



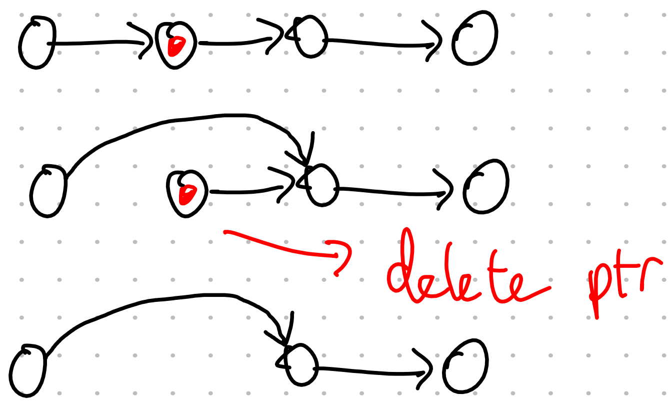
+ more resistant to mem. fragmentation
+ more efficient insertion, deletion, concatenation of 2 lists
- Need to traverse to get nth element

↓

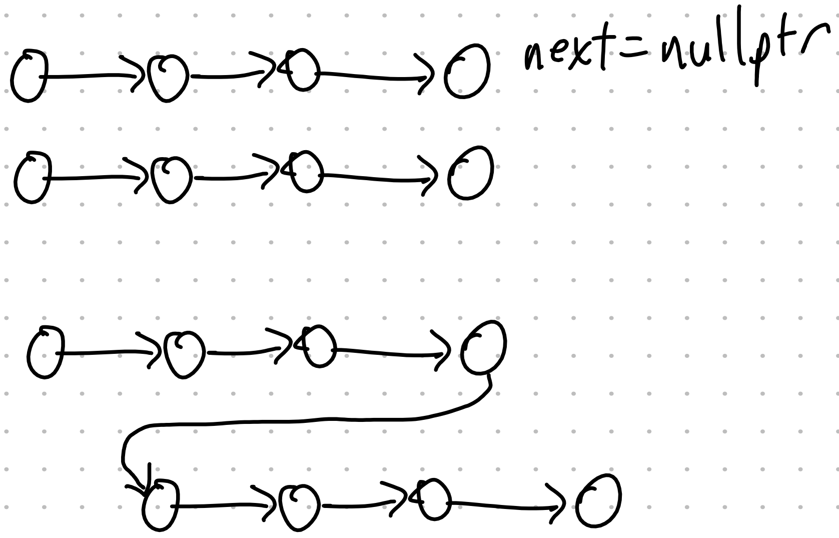
Insertion



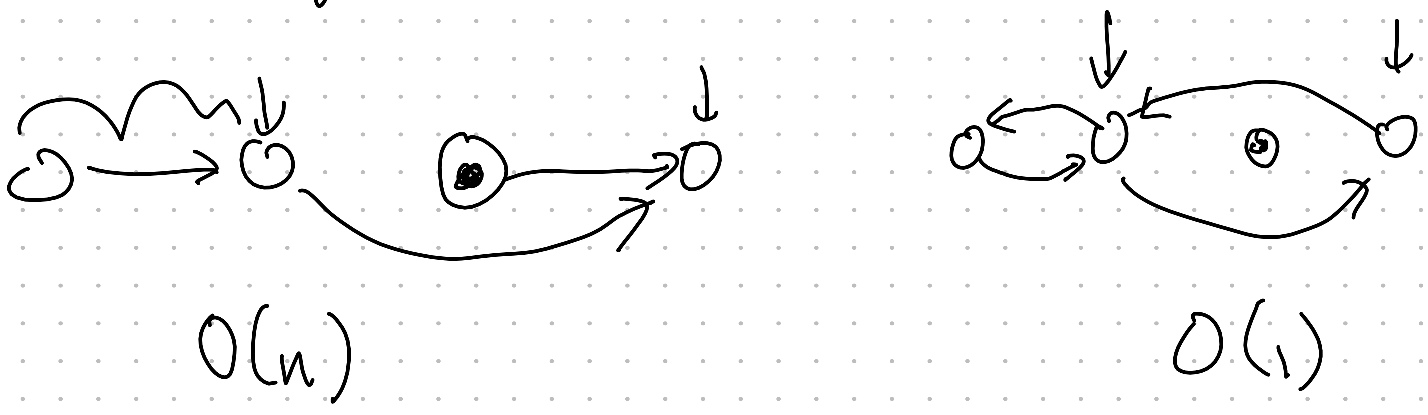
Deletion



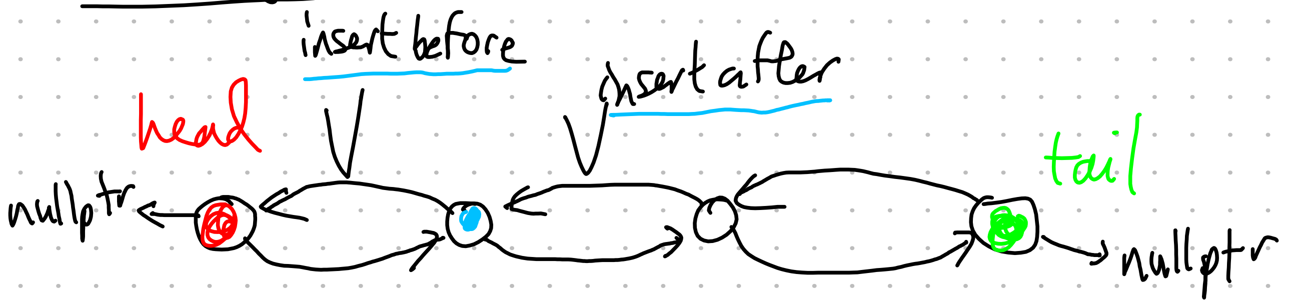
Concatenation



Complexity of deletion

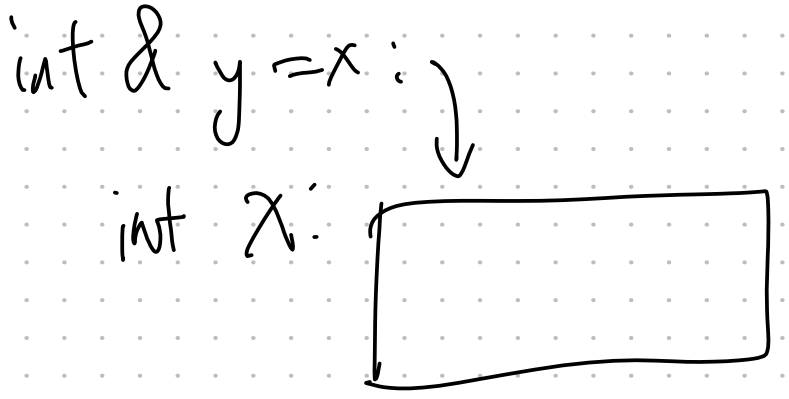


Doubly Linked Lists

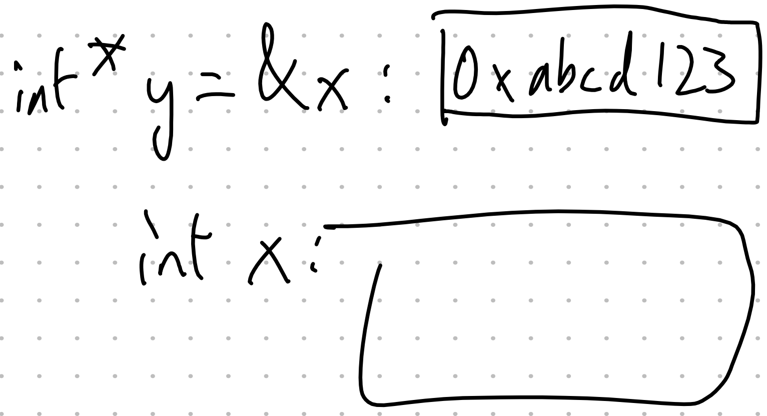


```
struct Node {
    int data;
    Node* next;
    Node* prev;
}
```

Additional capabilities
- traverse backwards
- insert before is easier
- delete is easier



```
int& y = x;
int x;
```



```
int* y = &x;  0xabcd123
int x;
```