Class → definition of user-defined data type Car

Objects → an instance / variable bucket of a class. Car c1;

- data member
- member functions / methods

- private → not visible outside class.
- public

access specifiers

security purpose

visible everywhere.

Special functions

① Constructors

- automatically called when an object is created.
- default
- copy Obj o1(o2);
- copy assignment Obj o1 = o2;

② Destructor

- automatically called when object is destroyed

- function overloading
  - you can define as many constructors you want, provided no two of them have same type of param seq.

Dynamic Allocation

↳ getting memory on-the-run from heap / free-memory

run-time

Vector as STL : features → able to increase size as needed

Overloading → of → function

we put the logic of MyVector into another file named MyVector.h and then use the definition there by include it.

+ → unary
+ → binary
+ → string

* → binary multi
* → pointer decl
* → dereference

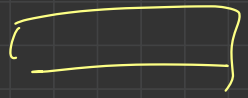
OOP
- abstraction → hiding away the actual implementation
- encapsulation
- polymorphism
- inheritence → deriving a class from another class

- function overloading is when two or more functions have same name but different parameters.
- Operator overloading:
  - we can't define new operators
  - we can't change the default/base function of the op.
  - one of the operands must be a UDT (object)

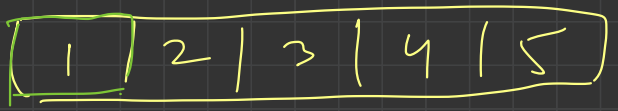
Array : fixed size, insertion deletion not so fast in arr

Linked List

- insertion
- deletion
- iteration

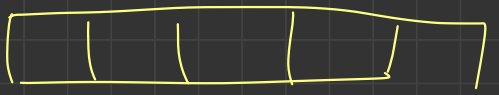


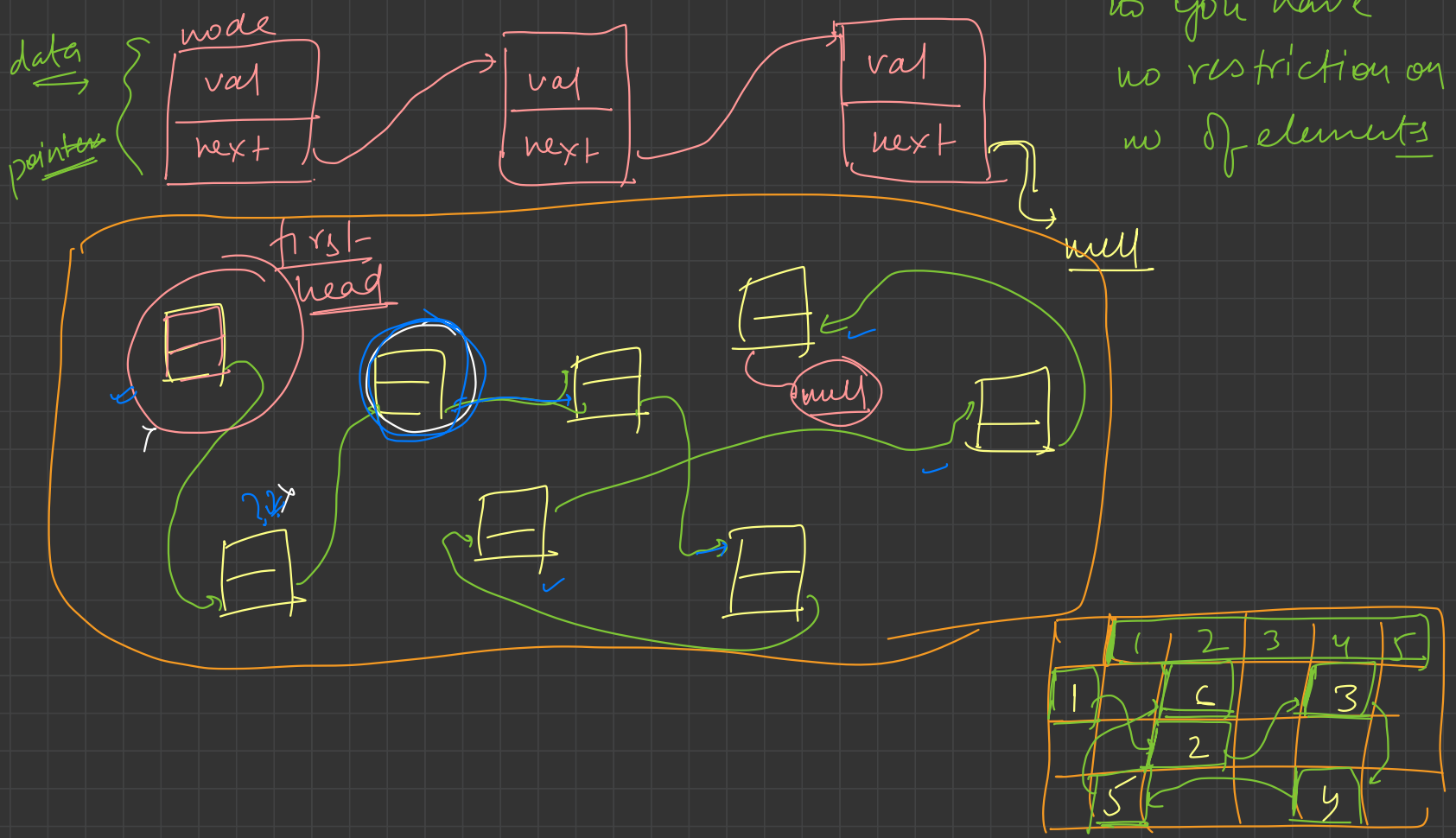
{1, 2, 3, 4, 5}



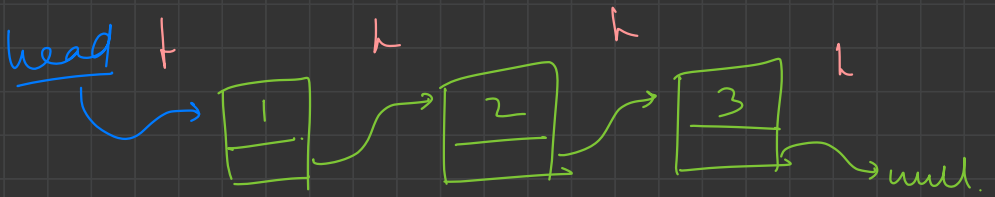
node → element/value

→ where the next element is stored





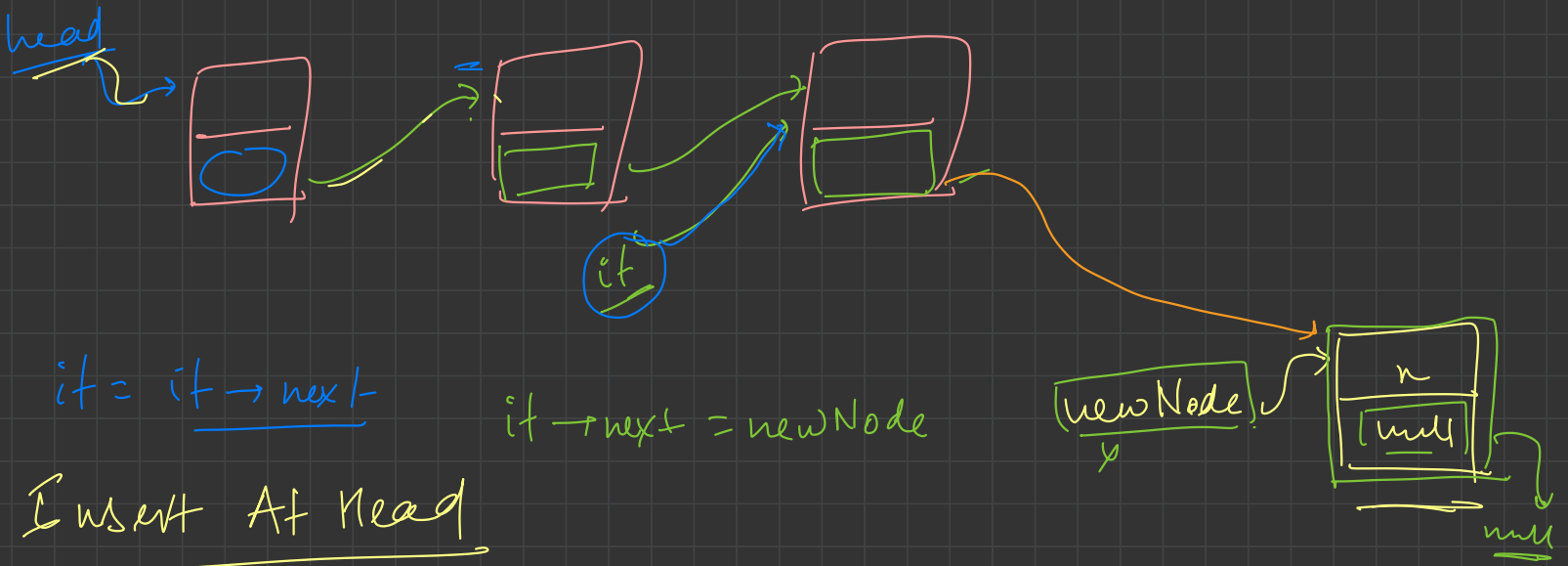
so you have no restriction on no of elements



insert (head, 4)

↳ has to create a node which has val=4 and needs to be attached somewhere in the list.

push_back → element inserted at last

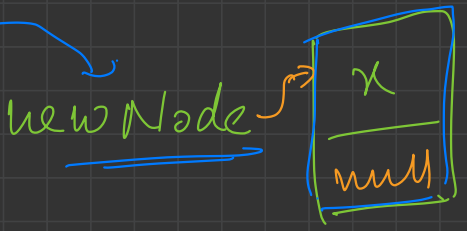


Insert At Head

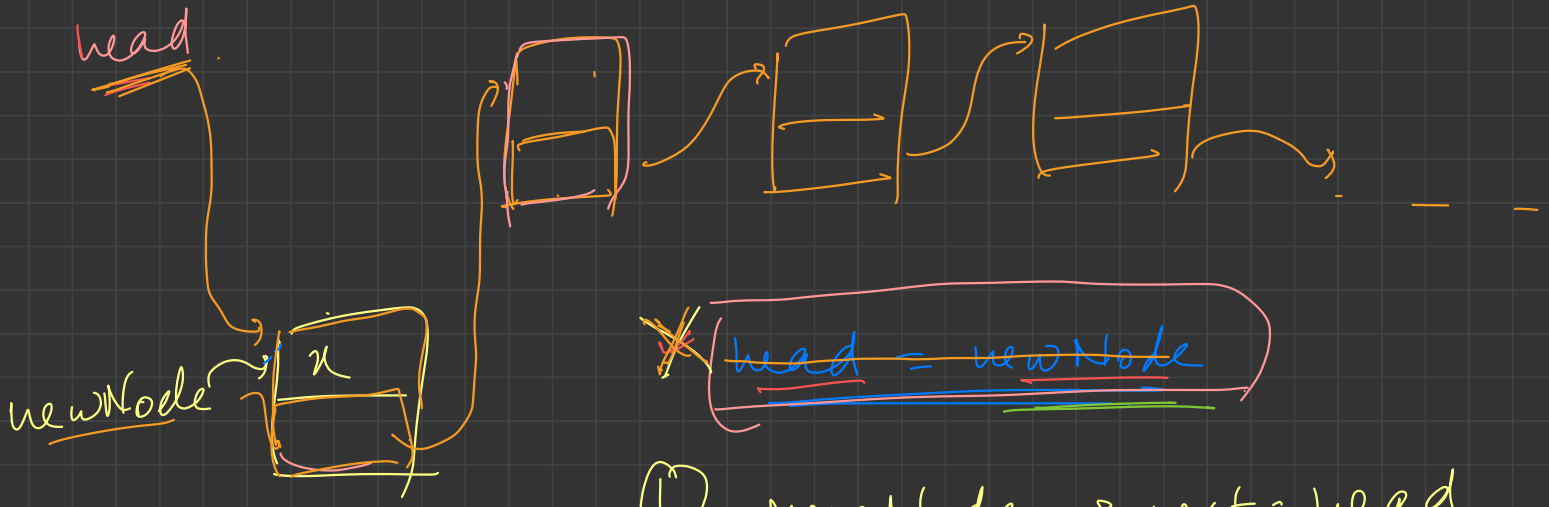
if head == null

head → null

head = newNode



Else head != null



① newNode → next = head

② head = newNode

① Write the entire code by yourself again
   ✓ by reference
   ✓ by return pointer/void

② write a function to find the length of the LL ✓

③ Insert at a given position ✓

④ find mid node ✓

⑤ Search a value in LL. ✓ & recursive

* If the LL has values in sorted order, what is the

best searching algorithm?

- Complexity of Binary Search on LL.