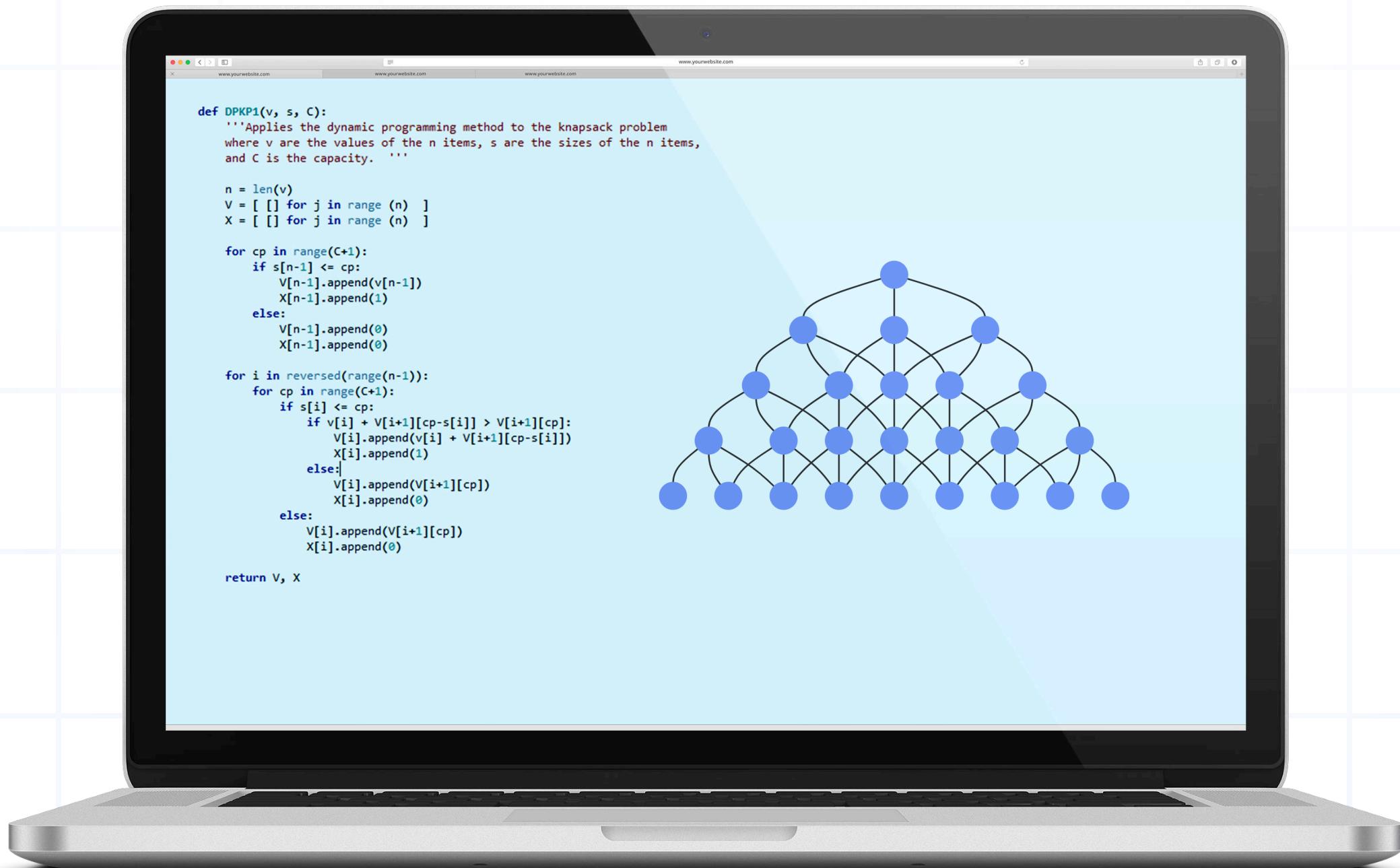


PATTERN RECOGNITION

LINKED LIST PROBLEMS



Pattern Recognition of Linked List Problems

Linked lists based problems are one of the most frequently asked in coding interviews.

It may feel overwhelming to cover all of the different types and patterns asked.

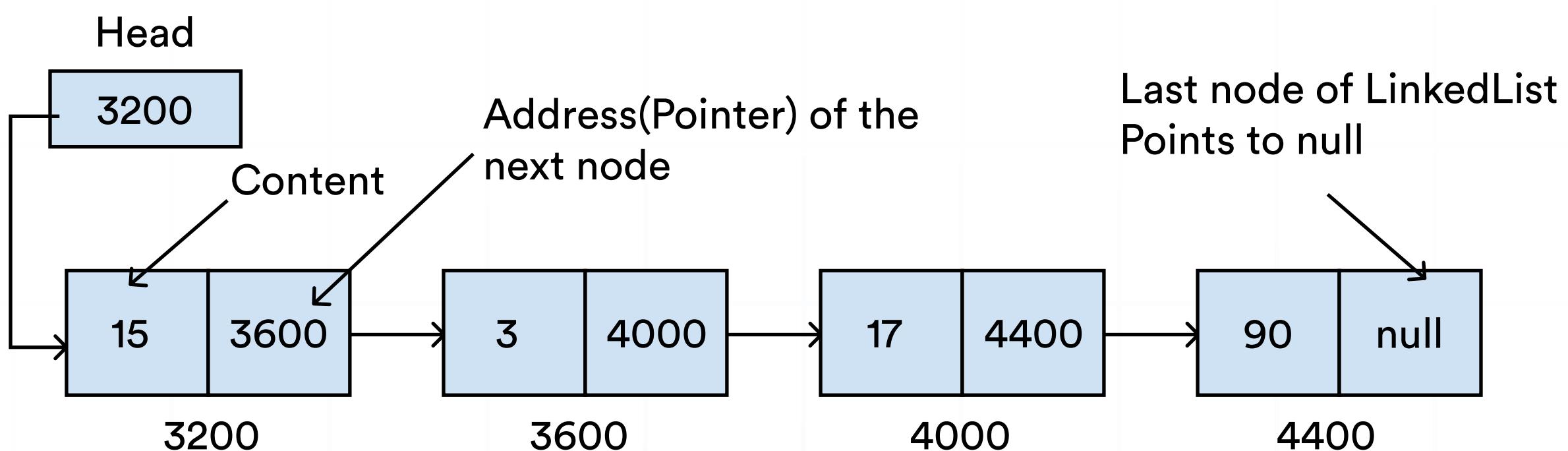
In this doc, we will look into different types of problems based on linked lists and how you should effectively approach to solve these problems.

Linked List

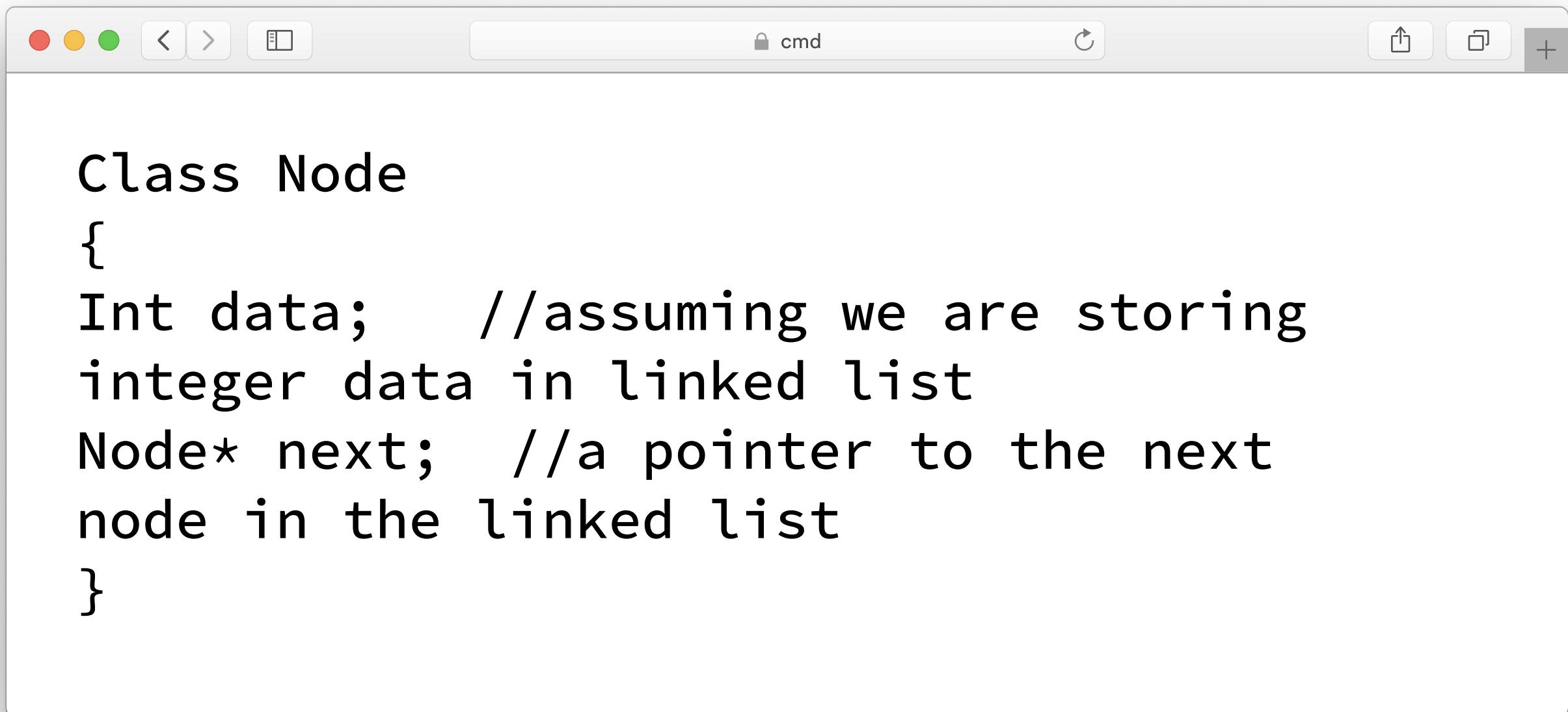
It is a linear collection of elements where the elements are **not stored in contiguous locations in the memory**.

Every element in a linked list, sometimes called a **node**, has two components- **data stored** and the **pointer to the next node**.

The first node is generally represented with a pointer called **HEAD**. A linked list is said to be terminated if the last node points to **NULL value**.



Basic Representation of a Linked List



```
Class Node
{
Int data;    //assuming we are storing
integer data in linked list
Node* next; //a pointer to the next
node in the linked list
}
```

Traversing a Linked List

In a linked list we traverse the elements by moving to the next element address according to the next pointers.

This is continued until we reach the NULL value, indicating that the linked list is terminated.

A general method for visiting all the elements of a linked list with 'head' node given:

```
Node *temp=head;
//create a temp pointer starting at head

while (temp != NULL)
{
    //continue the process until NULL value
    //is reached
    printf(temp->data); //current node data
    //value is printed
    temp = temp->next; //the temp pointer is
    //updated to reach the next node in the
    //linked list
}
```

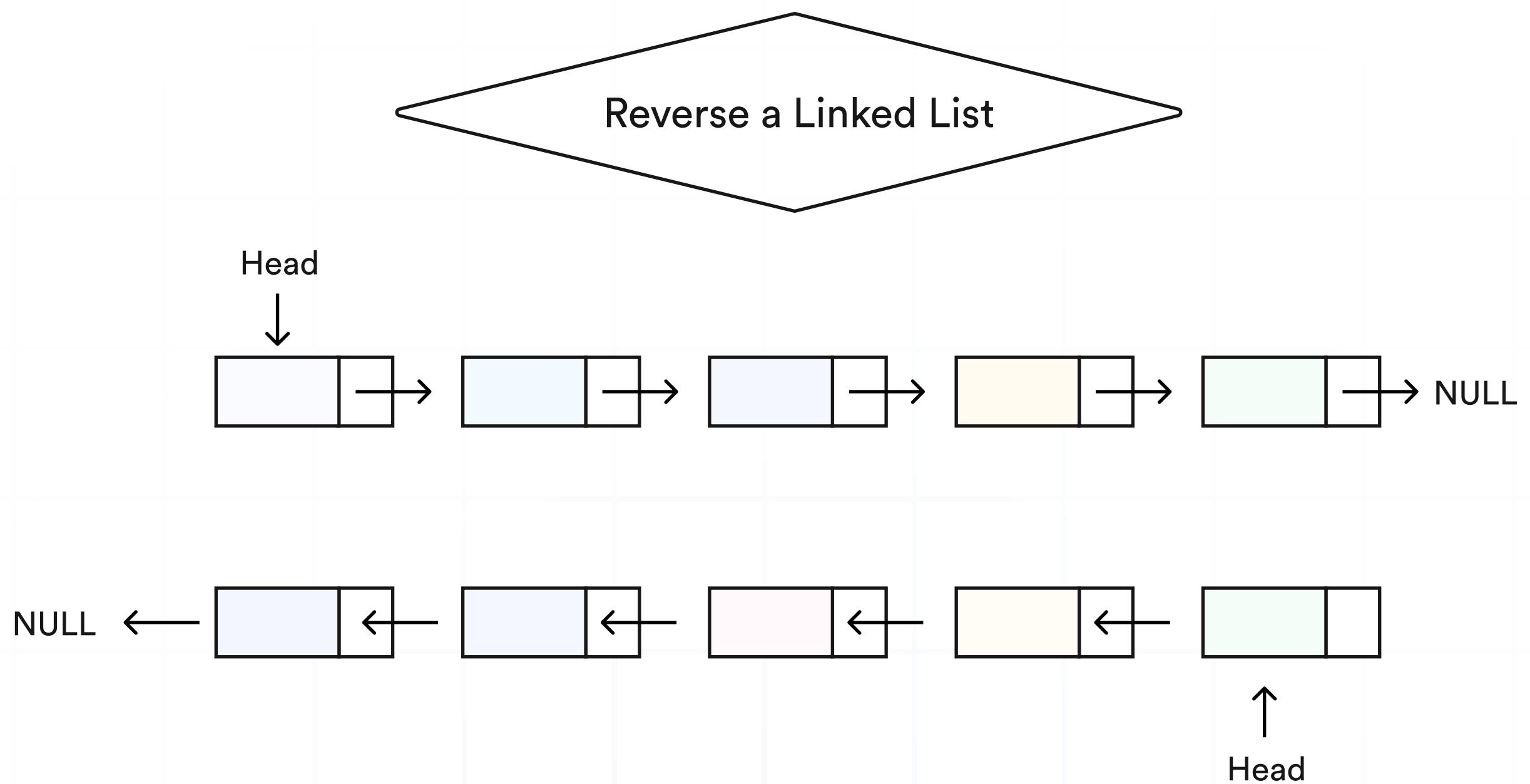
Now that our basics are mastered let's solve some questions!

Pattern 1: Three Pointer Approach

Reversing a Linked List:

Given a pointer to the first node or head of the linked list we need to reverse the links between each of the nodes.

This question is frequently asked by many of the top product based companies like Microsoft, Intuit, Adobe!



We can solve this problem by two ways, in both recursive and iterative methods.

Iterative Method

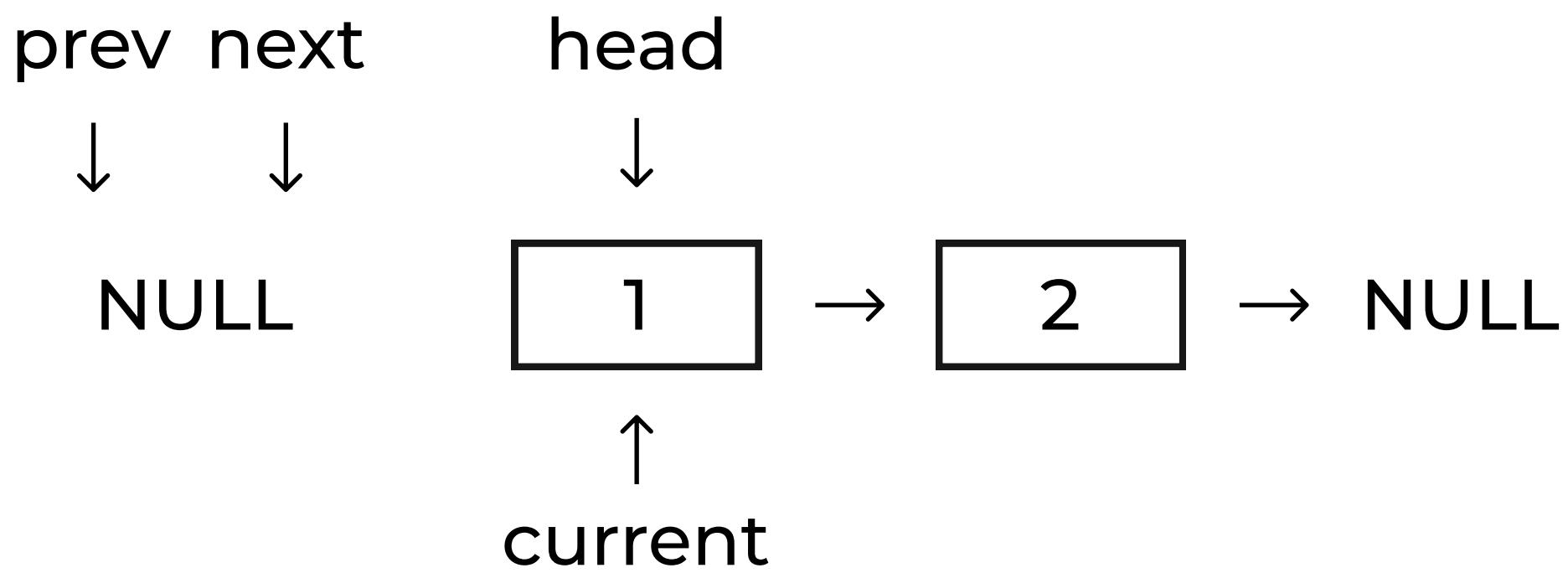
The iterative method uses three pointers: current is initialised to head, prev and next pointers are initialised to NULL.

We traverse the linked list until the current pointer reaches NULL and do the following in each iteration:

```
// Store the next node
    next = current->next;
// reverse the current node
    current->next = prev;

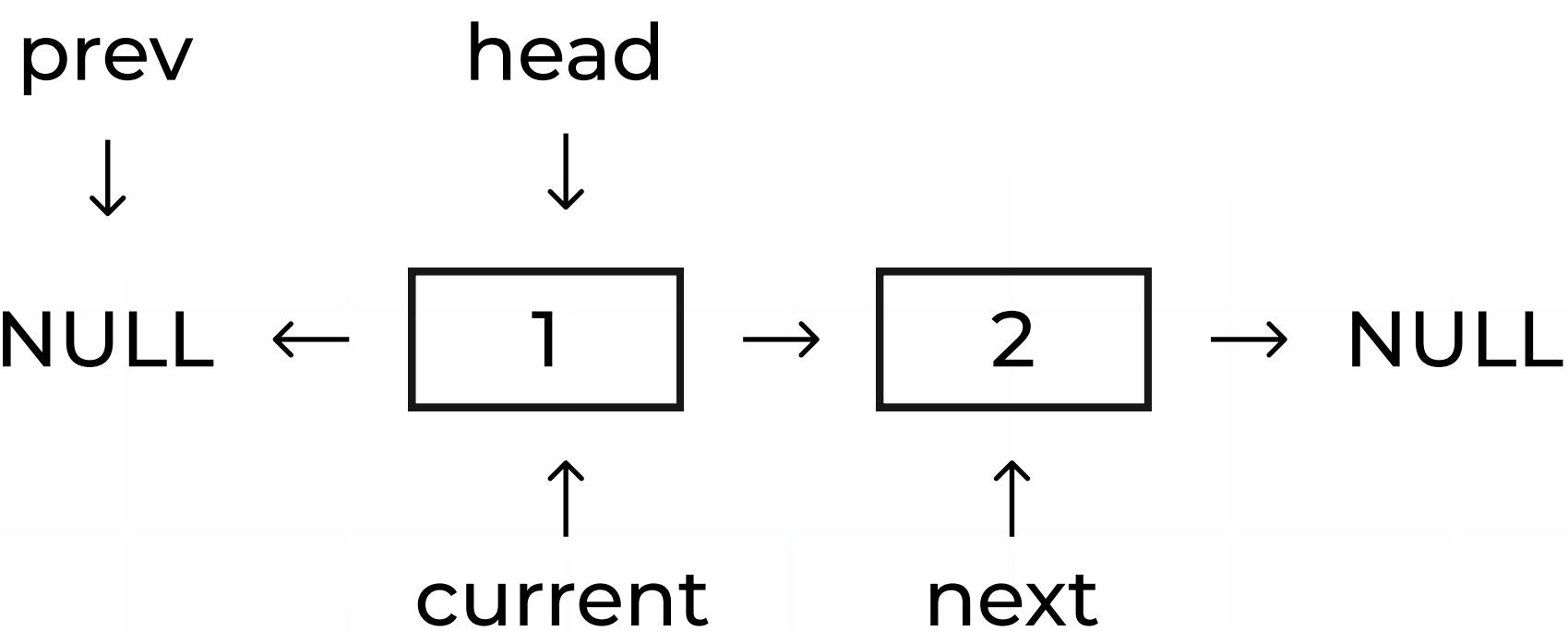
// move the pointers one step ahead in the
list
    prev = current;
    current = next;
```

①



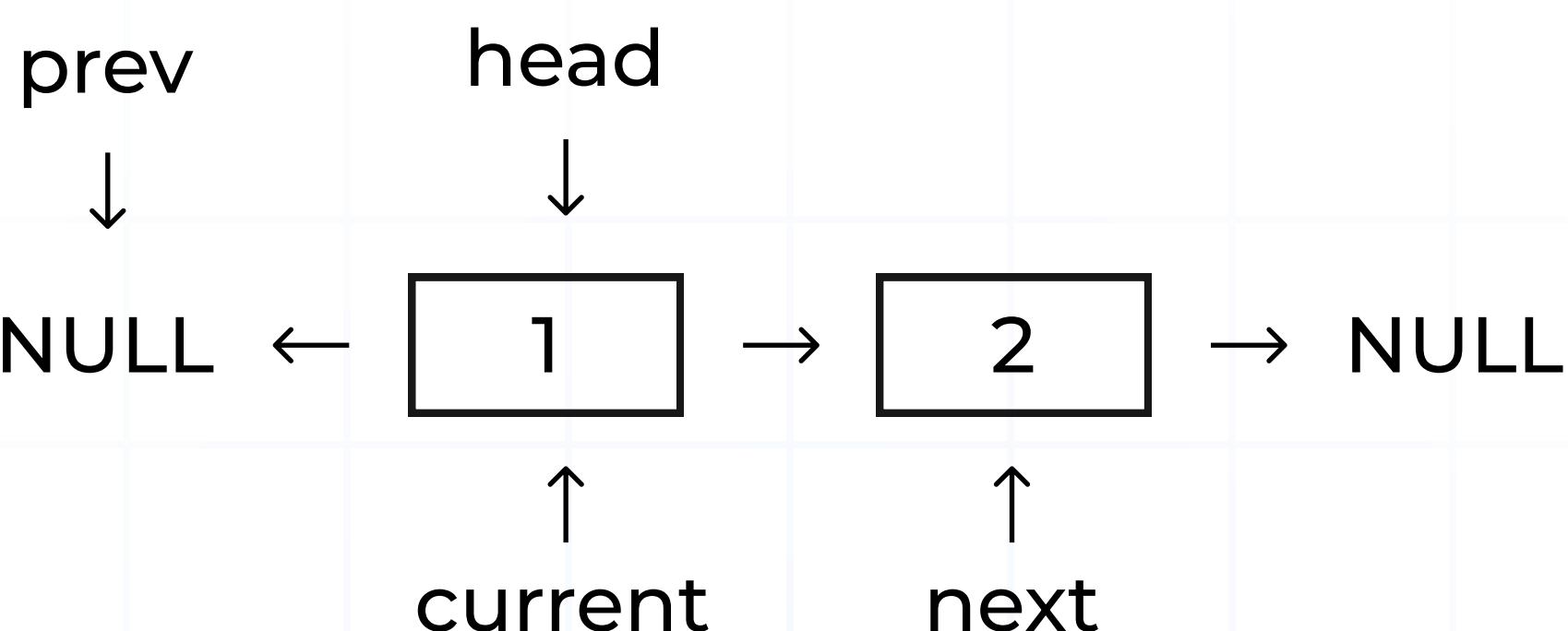
②

next = current → next



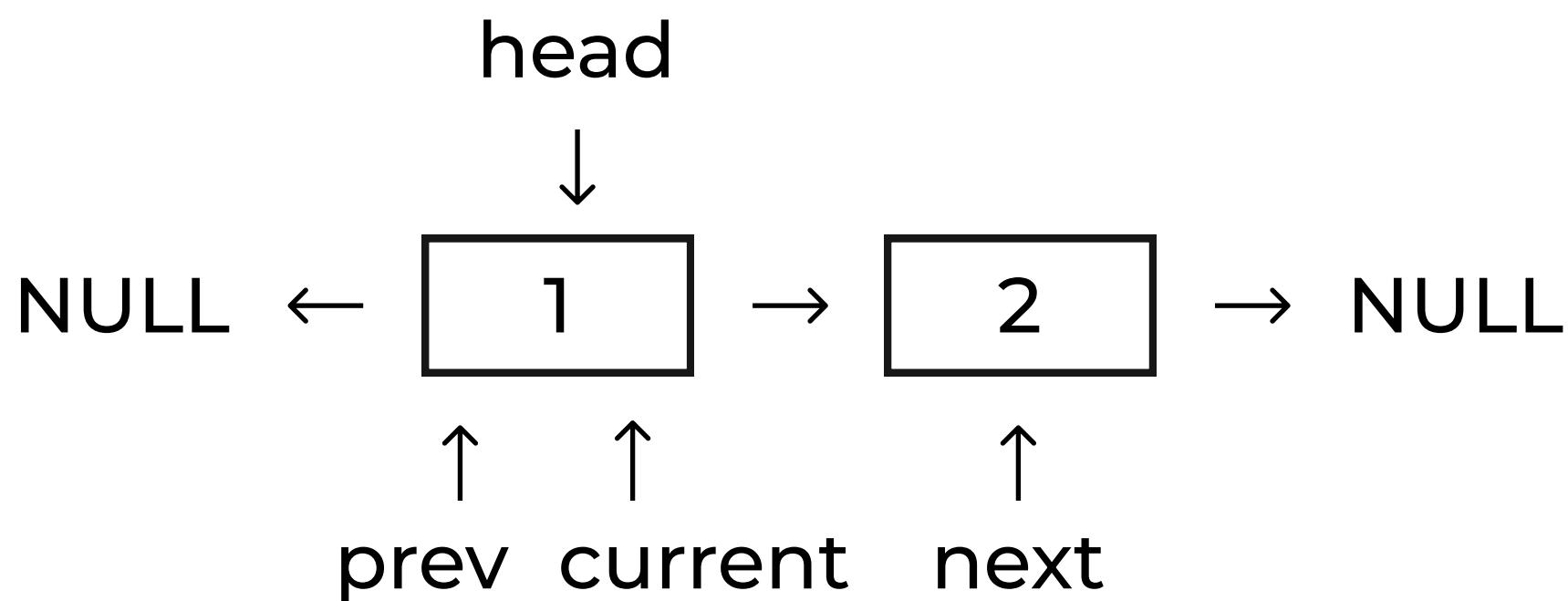
③

current → next = prev



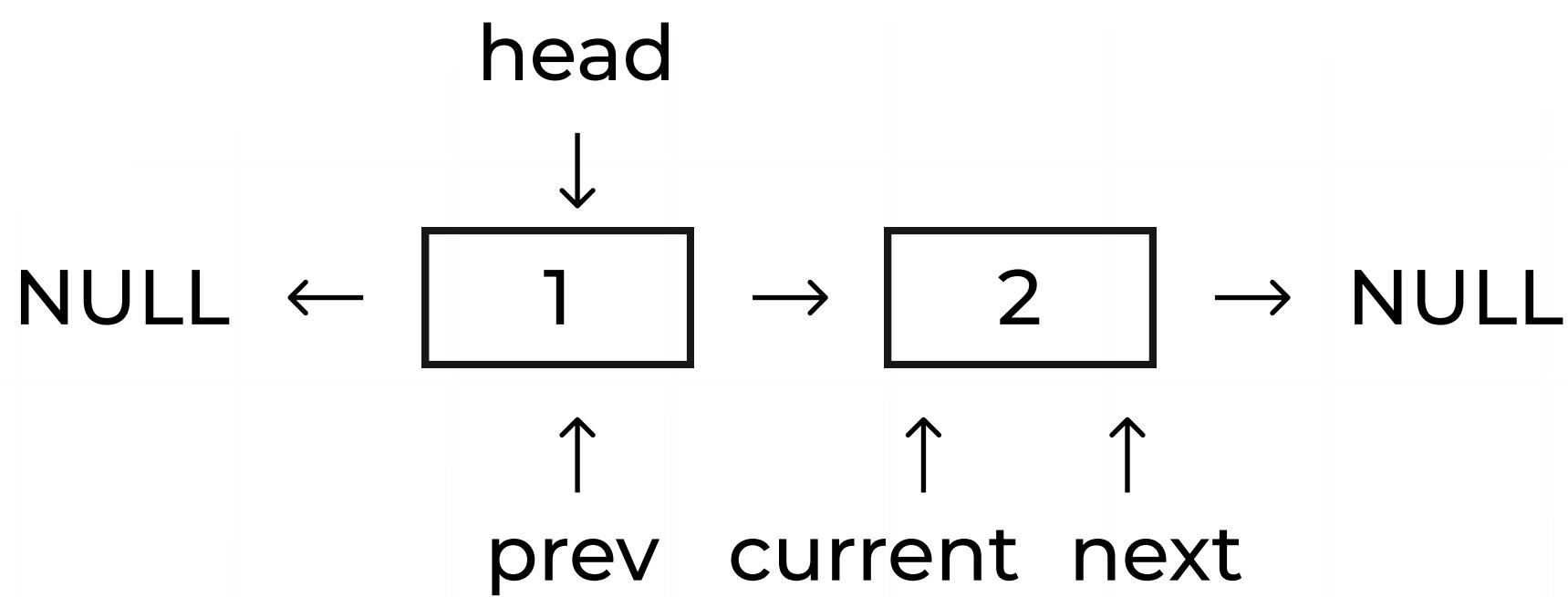
④

next = current



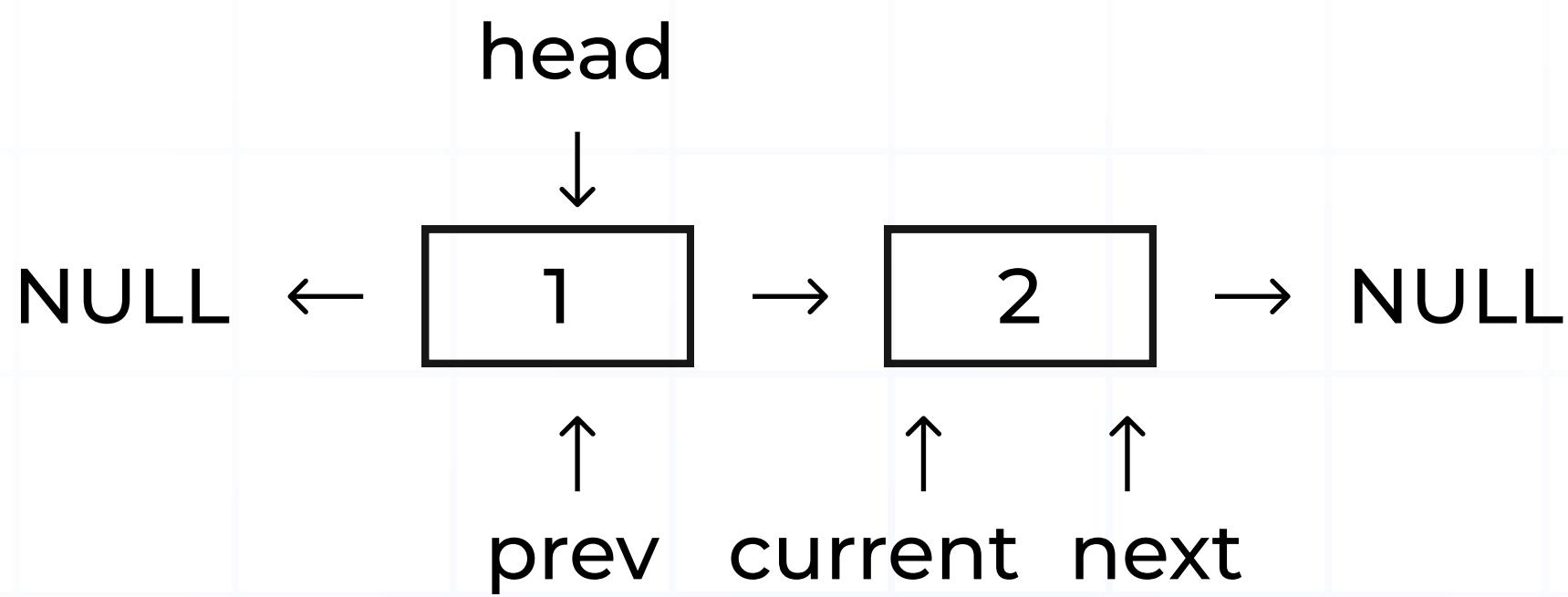
⑤

current = next



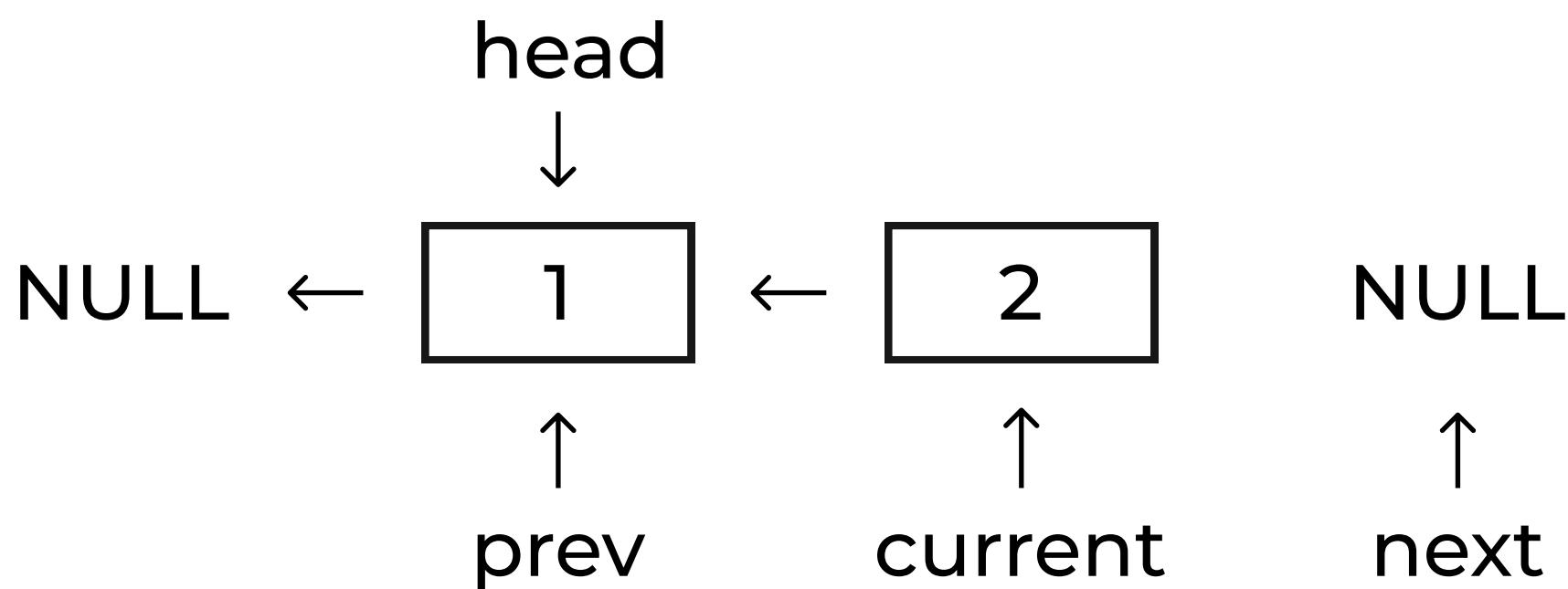
⑥

current = next



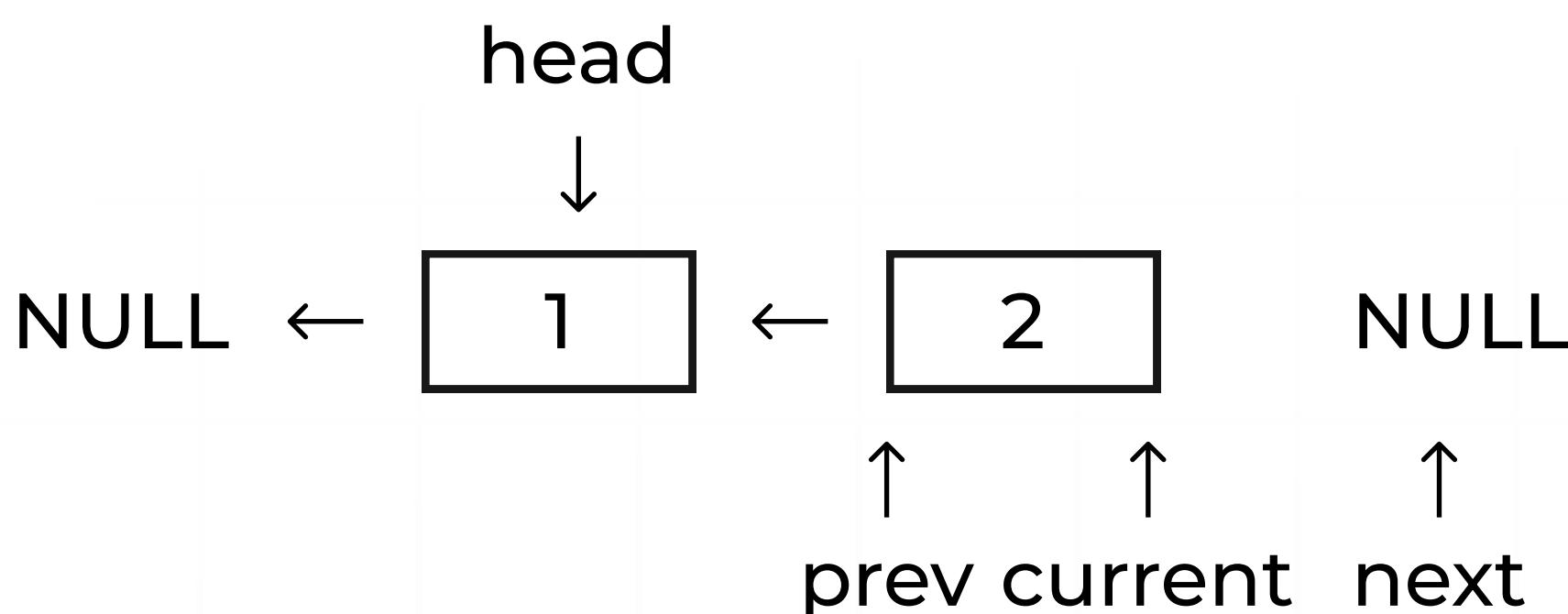
7

`current → next = prev`



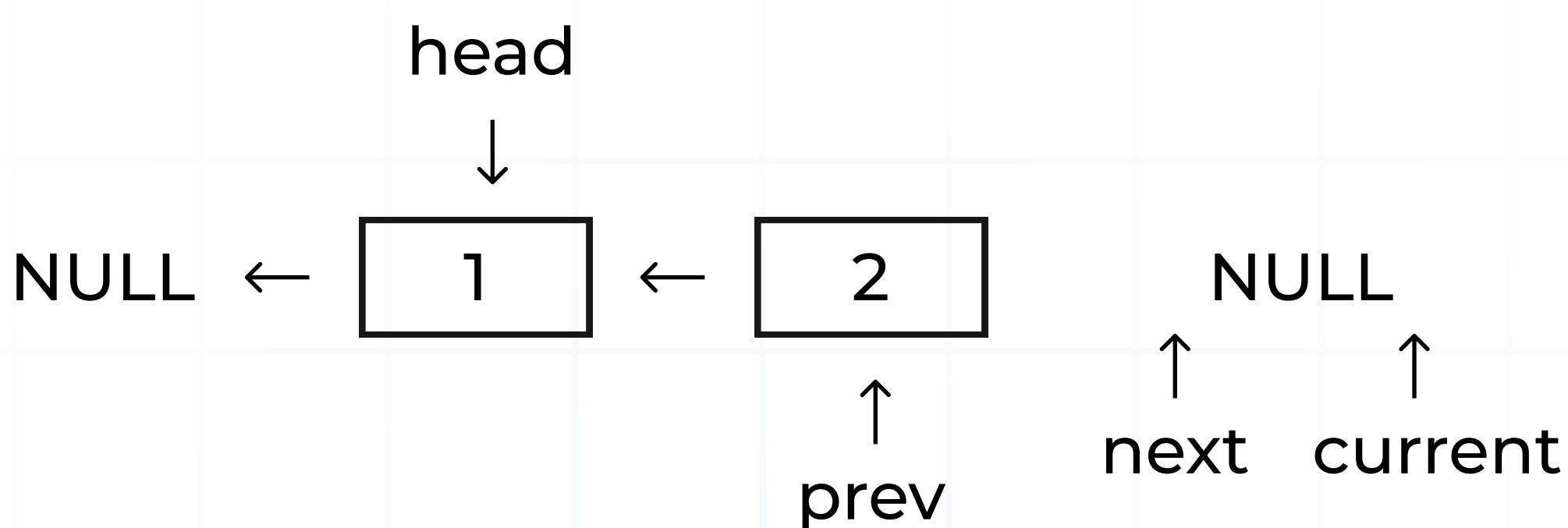
8

`prev = current`



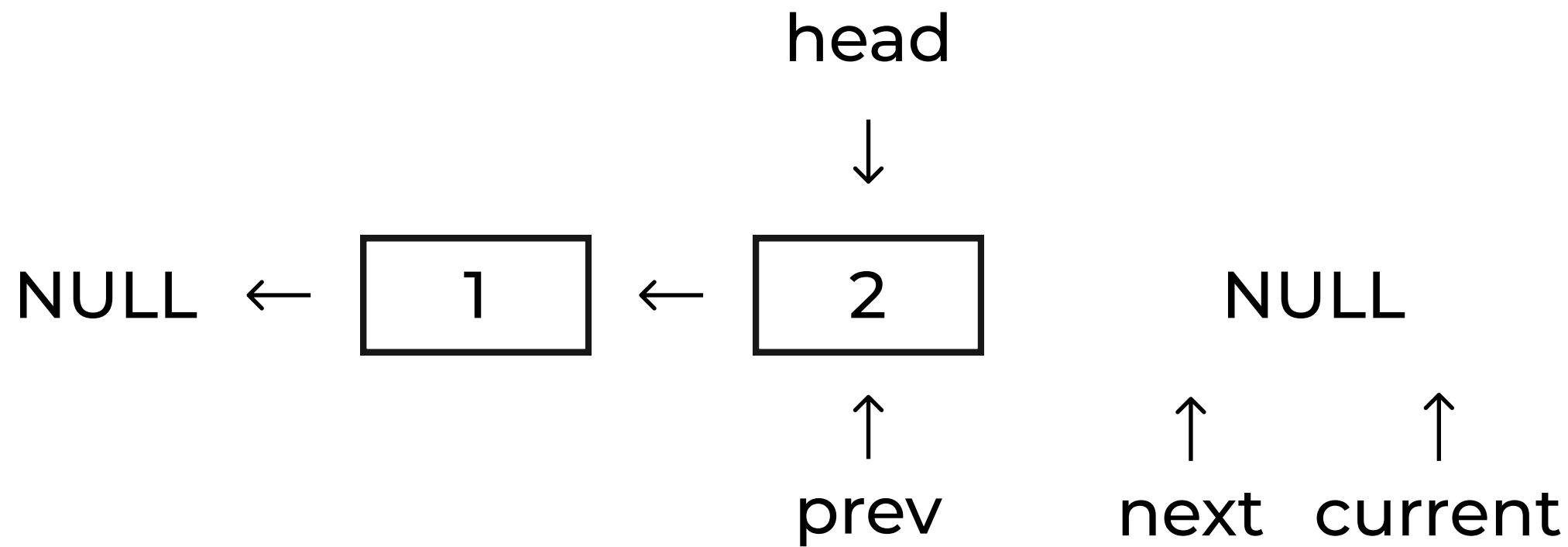
9

`current = next`



⑩

head = prev



The time complexity of this approach is linear as we will traverse through the list only once.

Time Complexity: $O(N)$ where N is the number of elements in the linked list.

Recursive Method

In this method we first divide the linked list in two parts – first node and remaining part of the linked list.

Then we call a recursive function to the remaining part until we reach the end of the linked list.

Once the rest of the linked list is reversed, the link between the first node and the rest of the linked list is reset.

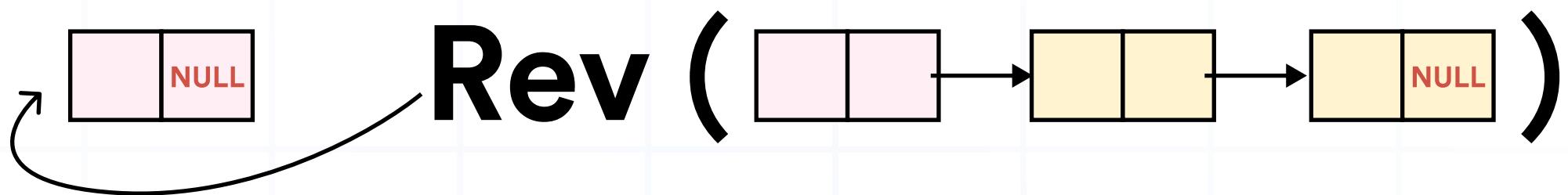
Finally the head pointer is updated to point to NULL.

Reverse a Singly Linked List

$$\text{Rev}(\text{NULL}) = \text{NULL}$$

$$\text{Rev}(\boxed{} \boxed{\text{NULL}}) = \boxed{} \boxed{\text{NULL}}$$

$$\text{Rev}(\boxed{} \boxed{} \rightarrow \boxed{} \boxed{} \rightarrow \boxed{} \boxed{} \rightarrow \boxed{} \boxed{\text{NULL}}) =$$



Code for Recursive Reversal of Linked List:

```
Node* reverse(Node* head)
{
if (head == NULL || head->next == NULL)
    return head;
Node* rest = reverse(head->next);
    head->next->next = head;
head->next = NULL;
return rest;
}
```

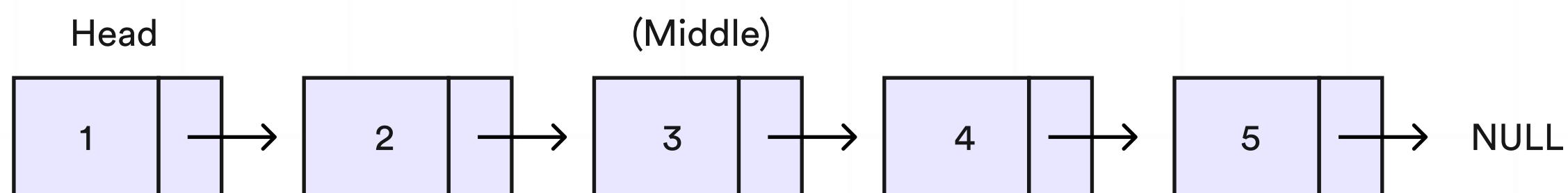
Pattern 2: Two Pointer Approach

Middle Element of Linked List :

Given a non-empty, singly linked list with head node head we have to return the middle node. If there are two middle nodes, we have to return the second middle node.

This is another famous question asked by many companies like Microsoft, Meta, Adobe.

A naive method would include two traversals of the linked list, one to find the total length ‘n’ and one to traverse till the ‘ $n/2$ th’ node and return it.



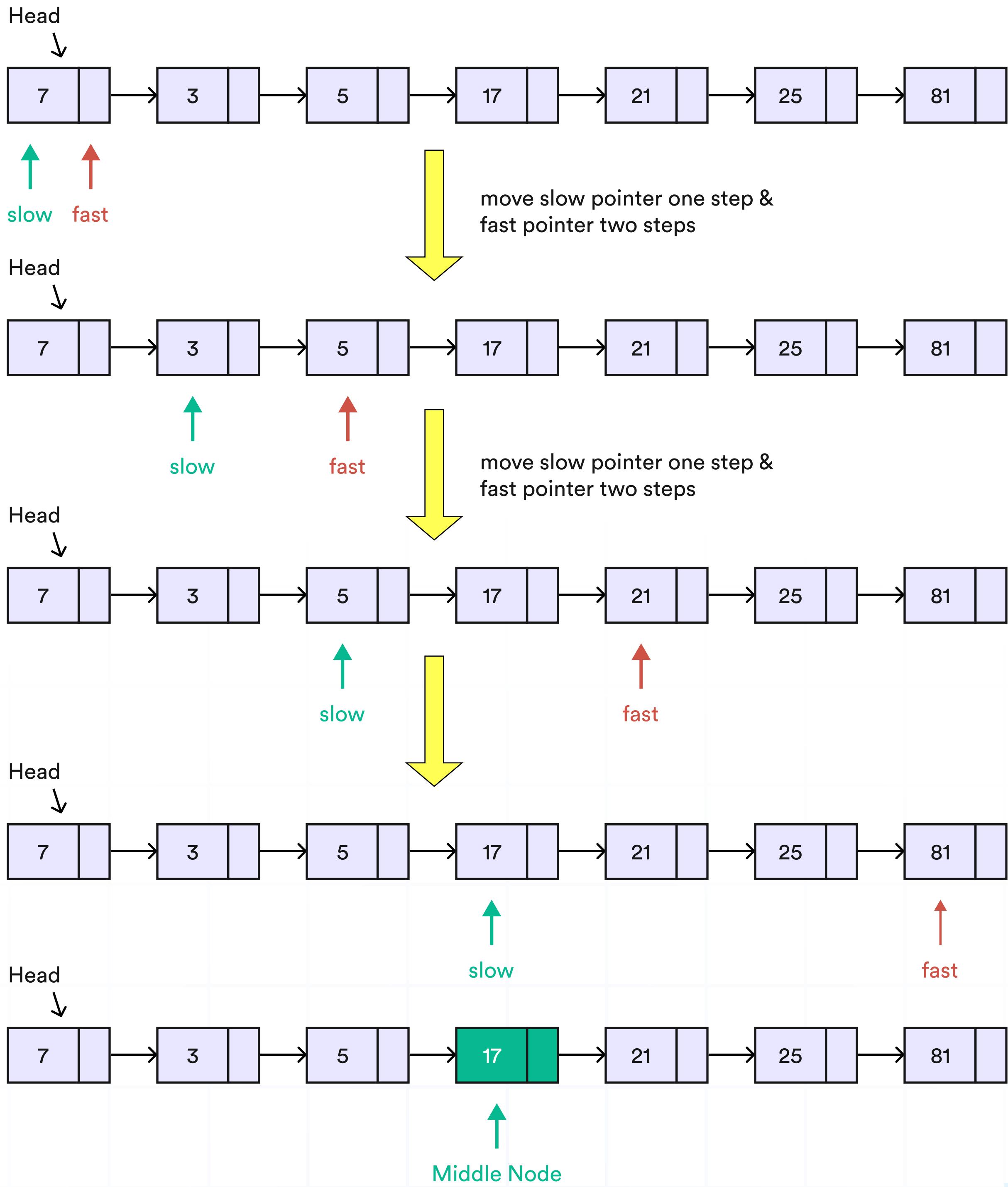
To solve this problem in a single traversal we use a famous method called the two pointers approach.

There are two pointers used in this method, slow and fast.

The logic is very simple, we traverse the linked list in two ways simultaneously.

1. First with the fast pointer we move forward by two steps in each iteration
2. With the slow pointer we move forward by only one step in each iteration

The beauty of this solution is that when the fast pointer reaches the end of the linked list, the slow pointer will have reached the middle of the linked list. Thus we can simply return the node that the slow pointer is pointing to.



Test yourself!

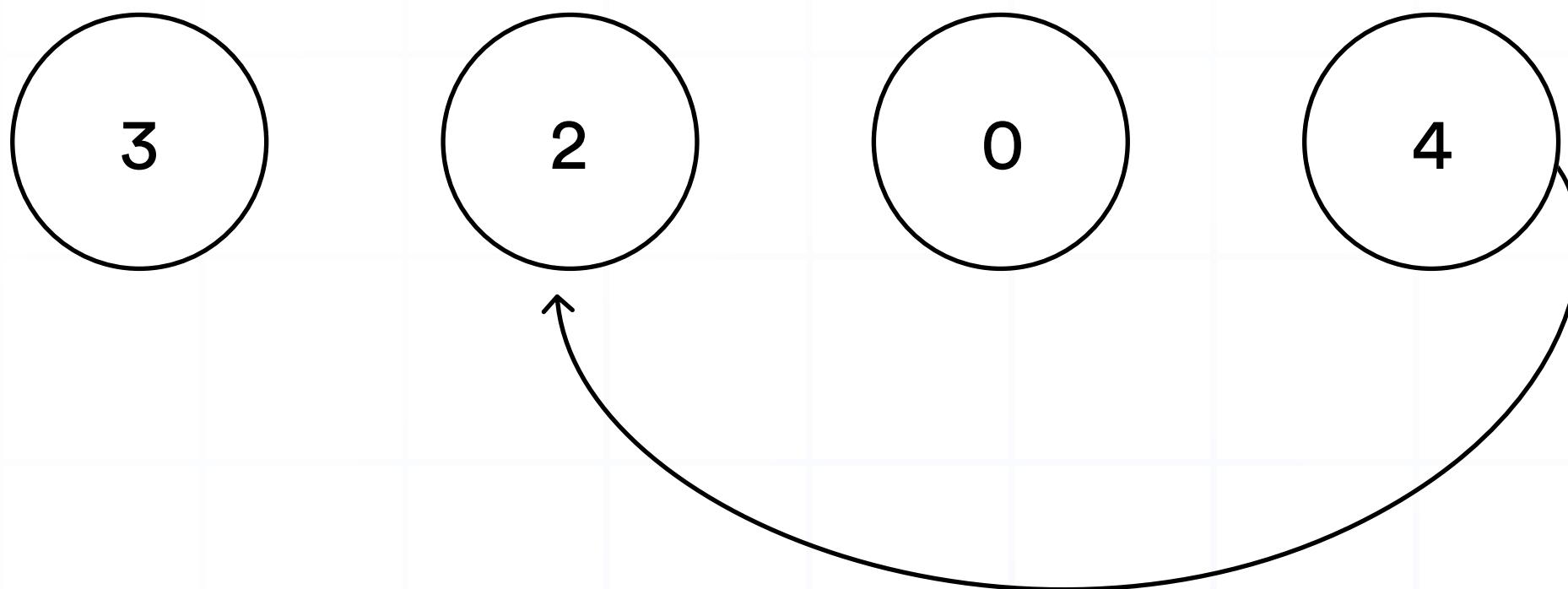
Now that you know how to approach linked list problems you can try this problem out.

Detect Cycle in Linked List

Given a linked list, return true if there is a cycle in the linked list.

A cycle in a linked list is a situation where a node that is visited already is visited again. Here is an example of a cycle in a linked list.

(💡 Hint: Look into if you can use the Slow and Fast Pointer approach discussed previously)



Other Important Linked List Questions:

1. Palindrome Linked List
2. Add numbers represented as linked list
3. Merge two sorted linked lists
4. Add 1 to a number represented as linked list
5. Remove Nth Node From End of List
6. Reorder Linked List
7. Sort a k sorted doubly linked list
8. Reverse a Linked List in groups of given size
9. Remove duplicate elements from sorted linked list
10. Reverse alternate k node in a singly linked list

ABOUT BOSSCODER

Bosscoder is an online upskilling platform for techies. We help learners upskill in tech roles to get them placed at top tech companies. We do so through our structured & mentored program designed by industry experts.

USP of our program include:

✨ **STRUCTURED CURRICULUM:**

Covers everything you need to get placed at top tech companies: Problem solving in DS & Algo, CS Fundamentals, System Design (HLD + LLD), Full stack Projects

✨ **LIVE CLASSES:**

An active learning classroom program taught by engineers working at companies like Microsoft, PayPal, Amazon

✨ **1:1 MENTORSHIP & MOCK INTERVIEWS:**

Personal mentors from top tech companies help you provide the right guidance, feedback, and support.

✨ **24/7 DOUBT SUPPORT:**

Through our army of Teaching Assistants

✨ **INDUSTRY-RELEVANT PROJECTS:**

Full stack specialization with Industry-relevant projects

✨ **PLACEMENT SUPPORT:**

Providing opportunities to tech engineers in eminent startups & top tech companies.

BUILD YOUR CAREER WITH US

-  **400+** Alumni placed at Top Product-based companies.
-  Highest package of **86 LPA**
-  Average package of **22 LPA**.
-  Resume reviewed and interview scheduled for **1000+** students



Lakshmi susmitha
Software Engineer II, JP Morgan

Service Based to JP Morgan in 4 months

From a tier-3 college to working in service-based companies, my thirst to join a product-based company didn't go away. BossCoder helped me provide a very detailed path from coding to system design. The way of teaching, and 1:1 mentorship helped me a lot.

Before → After
IBM → **JP Morgan**
Application Engineer Software Engineer II



Dheeraj Barik
Software Engineer 2, Amazon

System Engineer at Service Based to SDE 2 at Amazon

Working in Infosys, I was looking for a platform to prepare for interviews of product-based companies. I found BossCoder has a highly structured program covering DSA, System Design etc. in detail. Top-quality instructors and mock interviews proved helpful for me.

Before → After
Infosys → **Amazon**
Systems Engineer SDE 2



Vishal Srivastava
Software Developer, Barclays

Service Based to London Based Bank

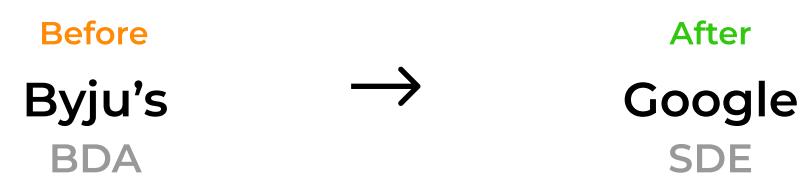
Doing self-prep, I couldn't even pass online assessments of Top companies. But the change BossCoder Academy brought into my preparation is phenomenal. Crucial topics taught in Live classes like DSA, HLD, and LLD, and my personal mentor's guidance ensured I clear my dream company.



Ujesh Nada
Software Development Engineer, Google

Business Development Associate to SDE at Google

I self-prepared DSA for 8 months, without any results. I joined BossCoder Academy since I wanted to be mentored by industry experts, and it proved to be a great decision for my DSA and System Design preparation. Their 1-on-1 mentor sessions and Live classes helped me transform my career.



Rakesh Kumar Satapathy
Sr. Developer, Hashedin

Bsc. Graduate stuck in service based to Hashedin

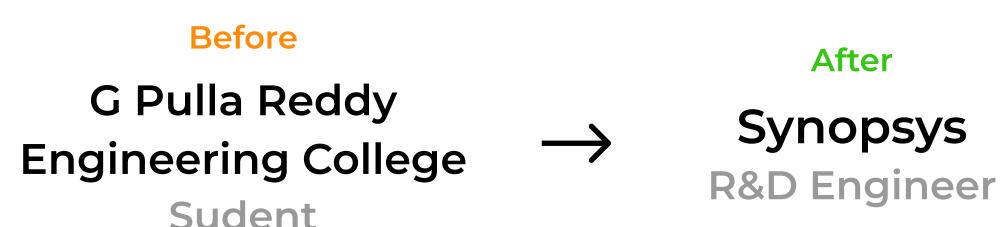
Stuck in a service-based company with no exposure, I always believed that I can realize my dream life, but didn't know how. BossCoder Academy showed the right path to coding geek inside me. Their world-class curriculum and personal mentorship enabled me to switch to my dream role.



Harshith Ravinoothala
R&D Engineer 1, Synopsys

Tier 3 College Student to Product Based Company

I always wanted to get into a product based company, but being a tier 3 college student lacked exposure to coding. BossCoder Academy helped me gain confidence in DSA and core subjects like OS, DBMS and System Design. Instant Mentor support helped me stay clear of doubts.

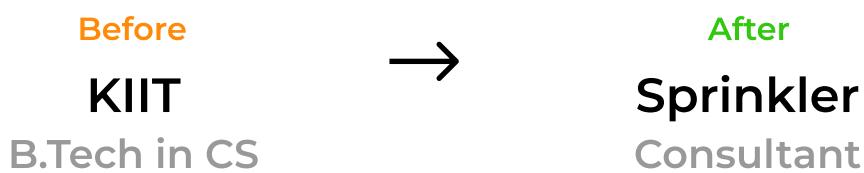




Sarveshwar Neogi
Consultant, Sprinkler

Clueless college student to Consultant at Sprinkler

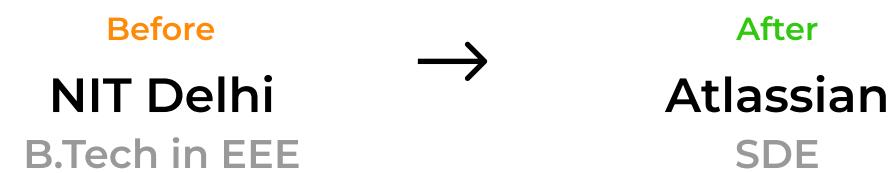
I was wasting my time in college, with no idea how to improve my coding skills. The structured roadmap provided by BossCoder transformed me into a Tech Rockstar. In-depth live lectures and daily handpicked questions helped me become consistent in problem-solving.



Aarushi Jain
Software Development Engineer, Atlassian

No interest in coding to SDE at Atlassian

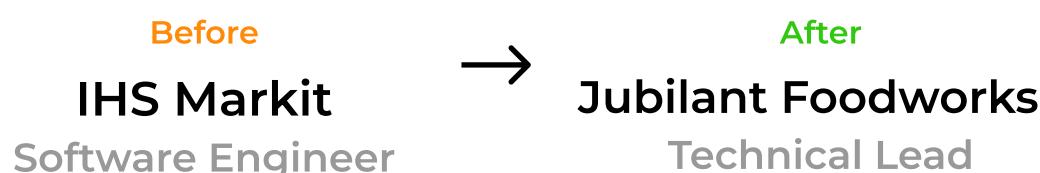
My journey in college was a roller coaster ride, and I wasted lots of time in learning from free resources. BossCoder Academy's excellent curriculum helped me become consistent and invest my efforts in the right direction. With my mentor's guidance, I received offers from Amazon and Atlassian.



Sumedha Khandelwal
Technical Lead at Jubilant Foodworks

Scared of Technical Interviews to Technical Lead

Having 7 years of experience, I always believed there is more to achieve in my career but failed in technical interviews. BossCoder helped me gain confidence to face technical interviews and I received many offers. Mock interviews and mentor feedback helped a lot.



Irshad K
Software Engineer, ByteDance

NIT Delhi to SDE in Singapore

I am among those talented students who require proper guidance to prosper. Cracking ByteDance was possible due to the guidance of my personal mentor at BossCoder Academy. Their structured curriculum helped me gain confidence in DSA and System Design.





Want to Upskill?

Go through our website, or email us at
ask@bosscoderacademy.com

VISIT WEBSITE
www.bosscoderacademy.com

A screenshot of a laptop displaying the BossCoder Academy website. The left sidebar has a dark blue background with white icons and text: Home, Payment, My Course (selected), My mentor, Placement, Calender, Store, Problems, Lead board, Refer Earn, Profile, and Support. The main content area is titled 'My Courses' and shows four course modules: 'Beginners Lectures' (100% completed, 20/20 problems solved, View Module), 'Advance DSA' (80% completed, 15/20 problems solved, View Module), 'High Level design' (0% completed, 00/20 problems solved, View Module), and 'Low level design' (0% completed, 00/20 problems solved, View Module). On the right side, there's a user profile for 'R. Ekunde' with a welcome message, and performance metrics: Points (1856, 15/1000), Streak (32, 32 Days), Solution (64, 64 / 100), and Rating (4.8, 4.8/5). At the bottom, there are buttons for 'Full course context' and 'Full report'.

BOSSCODER Academy

My Courses

All your courses will be available here.

Beginners Lectures ...

100% completed 20/20 Problems solved

View Module

Advance DSA ...

80% completed 15/20 Problems solved

View Module

High Level design ...

0% completed 00/20 Problems solved

View Module

Low level design ...

0% completed 00/20 Problems solved

View Module

R. Ekunde Welcome back

Points 1856 15/1000

Streak 32 32 Days

Solution 64 64 / 100

Rating 4.8 4.8/5

Full course context

Full report