CMSC 21

FUNDAMENTALS OF PROGRAMMING

Kristine Bernadette P. Pelaez

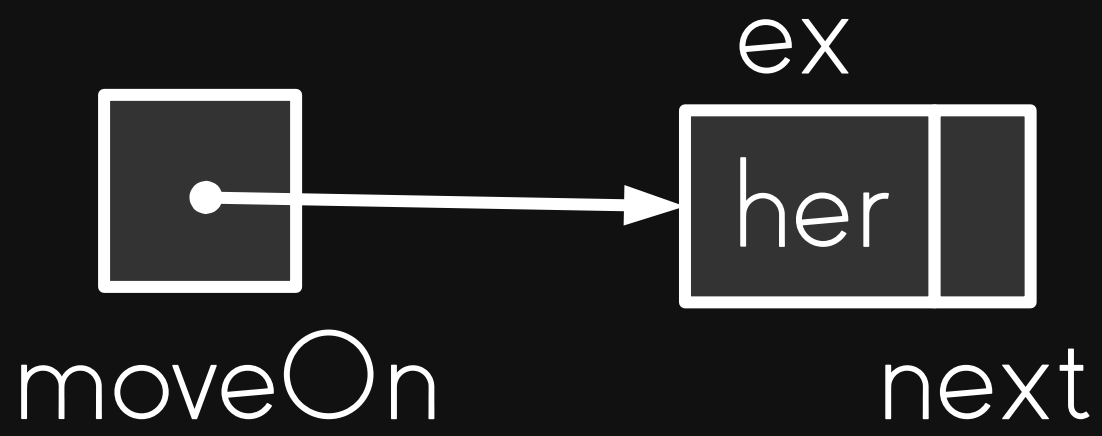
Institute of Computer Science
University of the Philippines Los Baños

DOUBLY LINKED LISTS

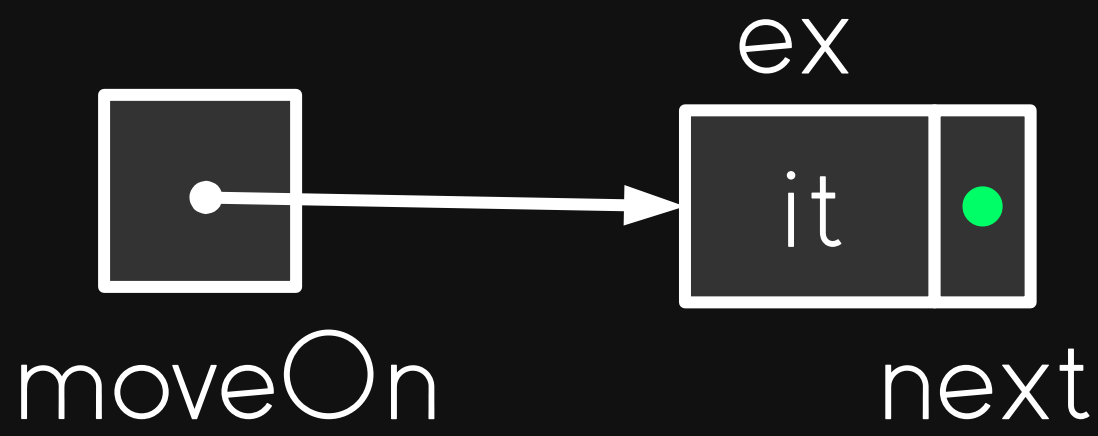
Quiz!

Define a self-referential structure that has one string member of size 64 named **ex** and a self-referential pointer named **next**.

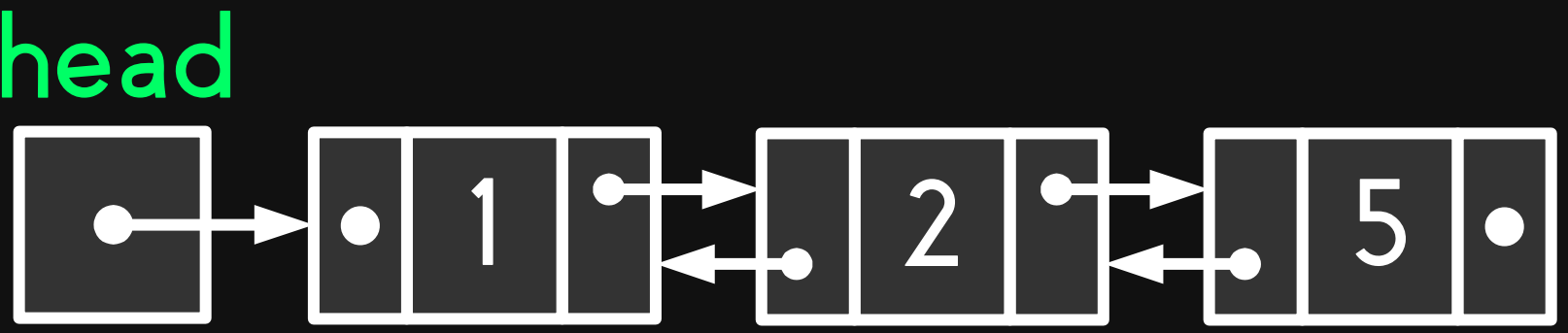
Declare a pointer named **moveOn** that will point to a node of the same type as the one previously defined.



Assume that the linked list being identified by **moveOn** is not empty, give the **next** pointer of the head node a value of **NULL**.

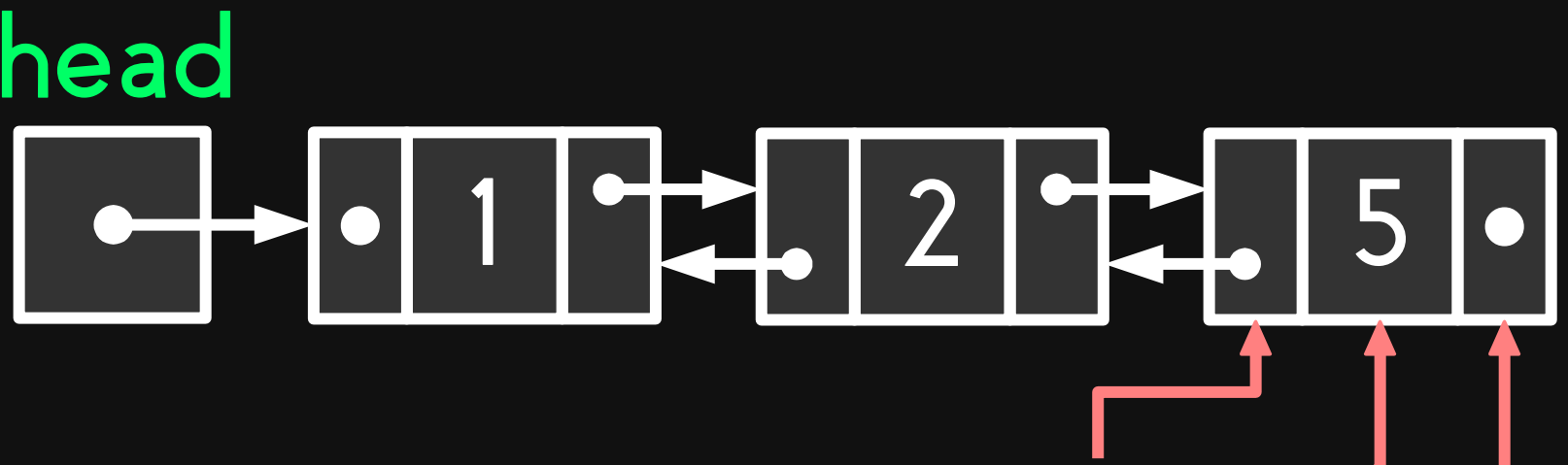


Doubly Linked Lists




```
struct NODE{
   struct NODE *prev;
   int num;
   struct NODE *next;
};
```

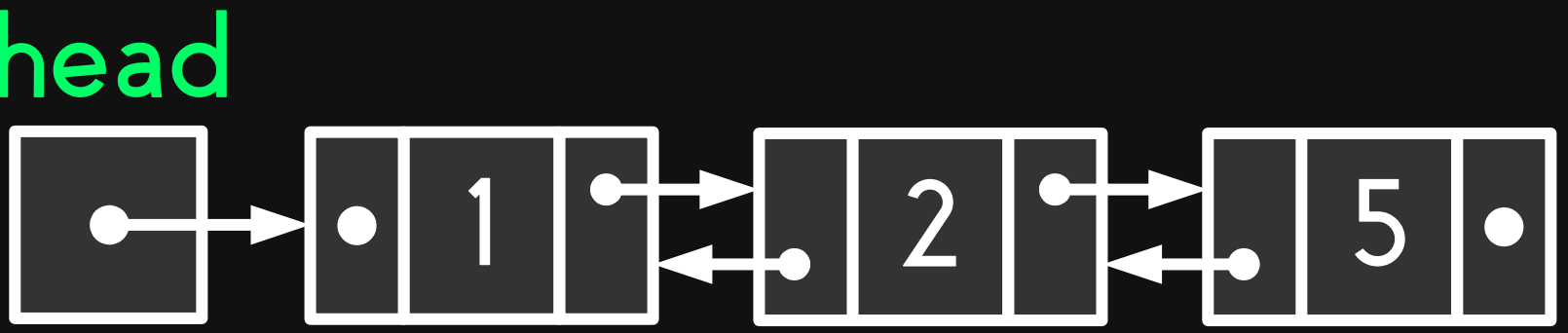
Doubly Linked Lists




```
struct NODE{
  struct NODE *prev;
  int num;
  struct NODE *next;
};
```

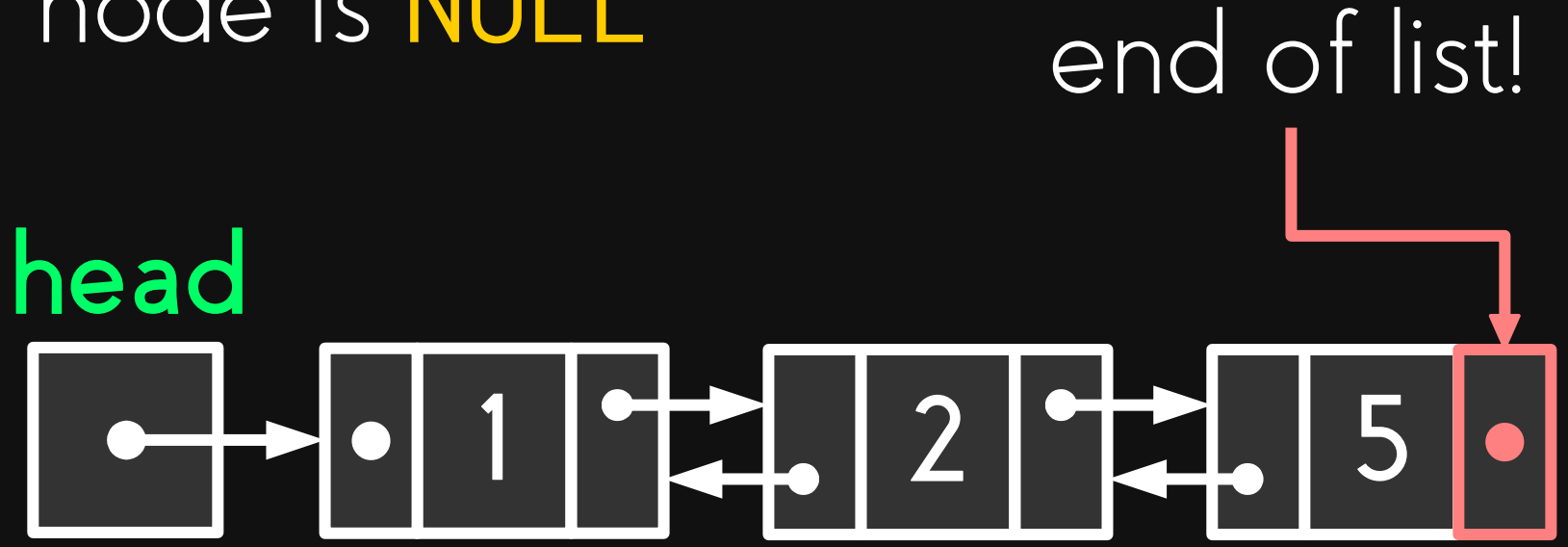
Doubly Linked Lists

the `next` of the
tail node is NULL



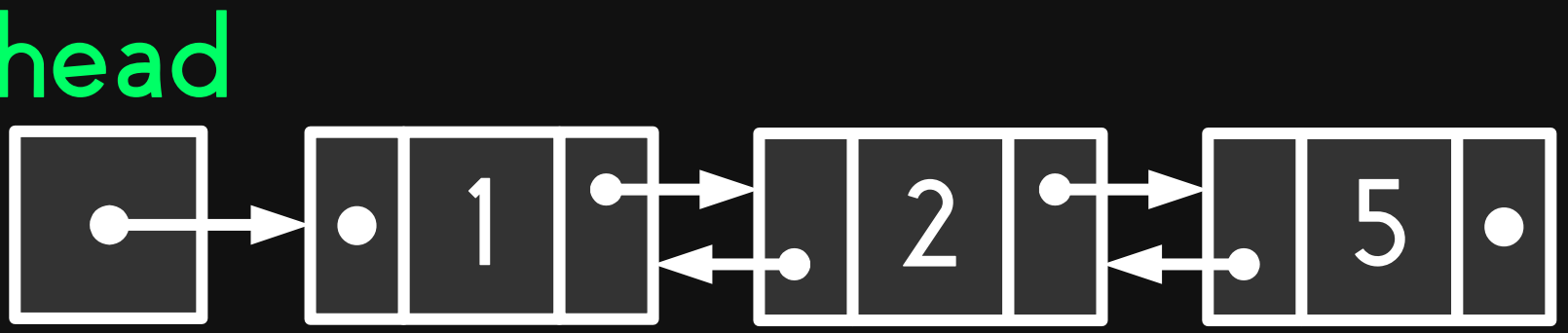
Doubly Linked Lists

the `next` of the
tail node is NULL



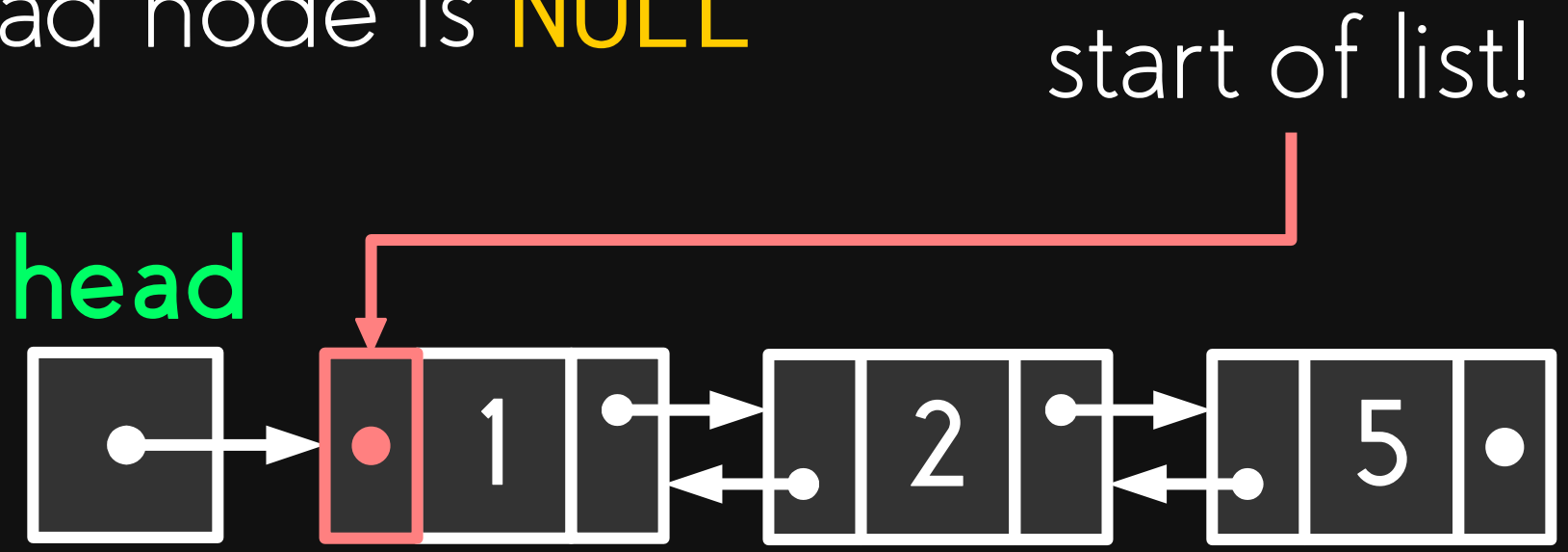
Doubly Linked Lists

the prev of the
head node is NULL

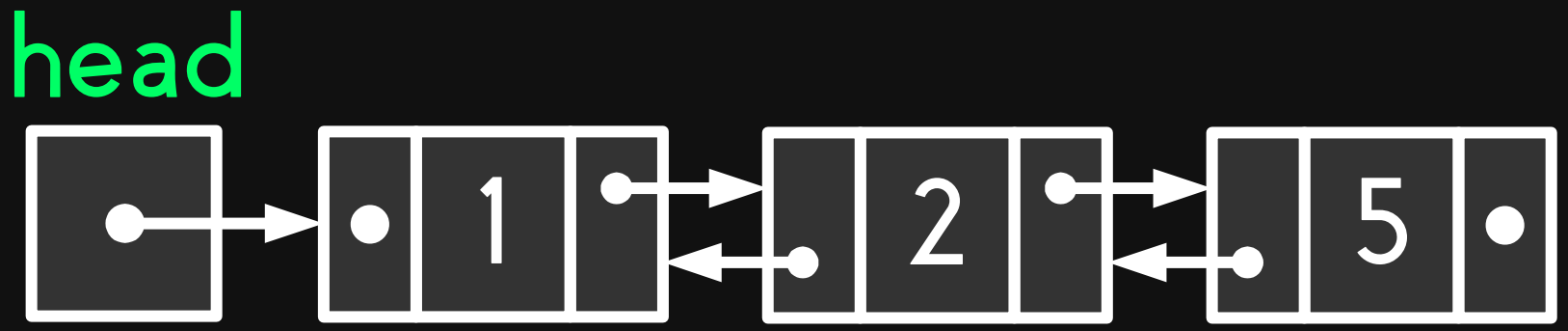


Doubly Linked Lists

the prev of the
head node is NULL

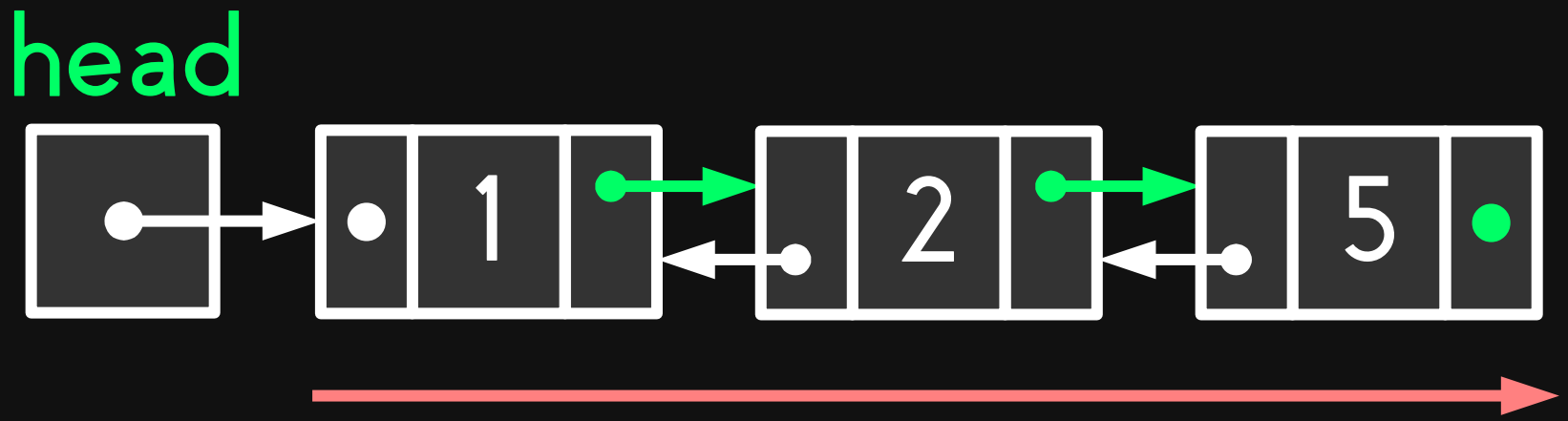


Doubly Linked Lists



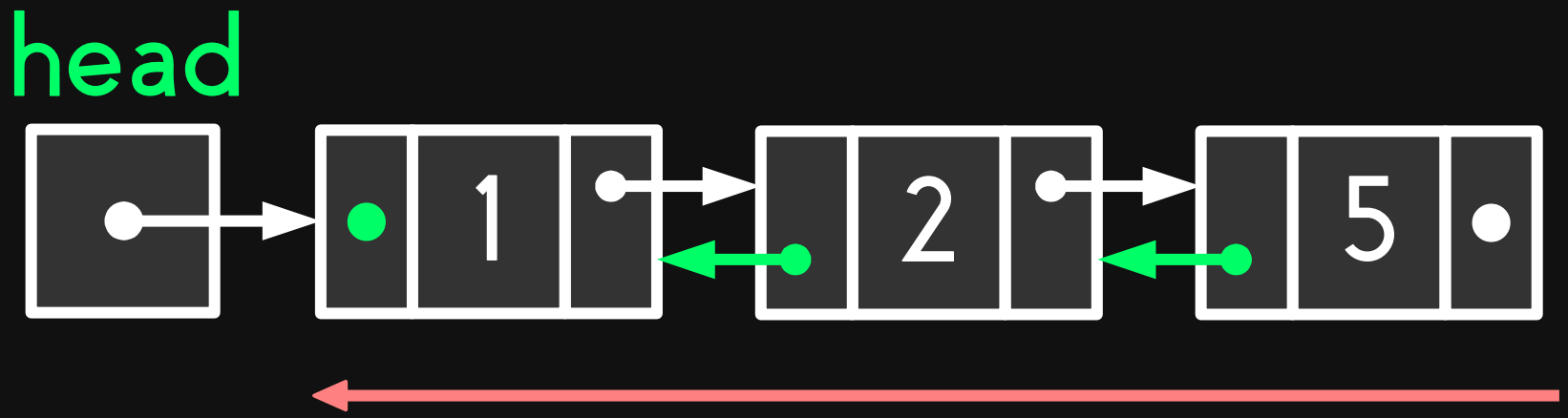
it allows you to go back!

Doubly Linked Lists



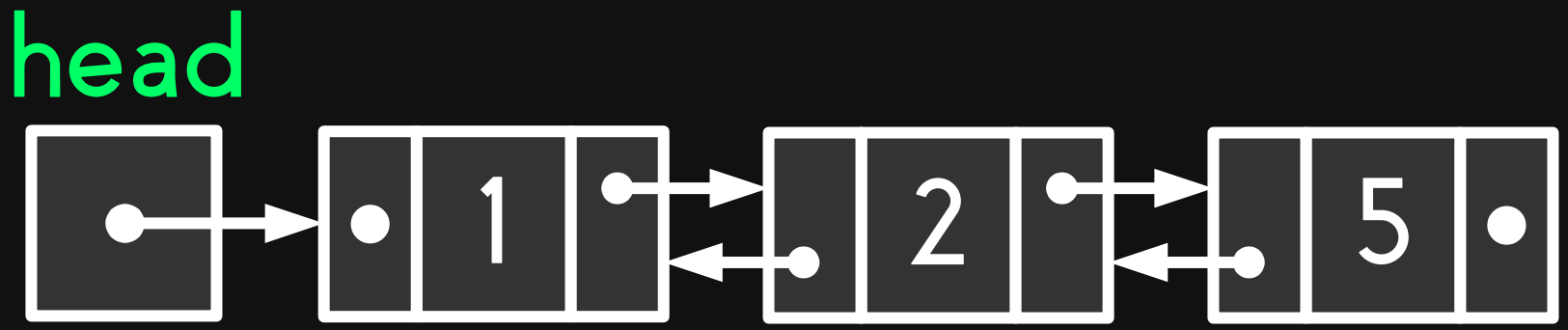
FORWARD

Doubly Linked Lists



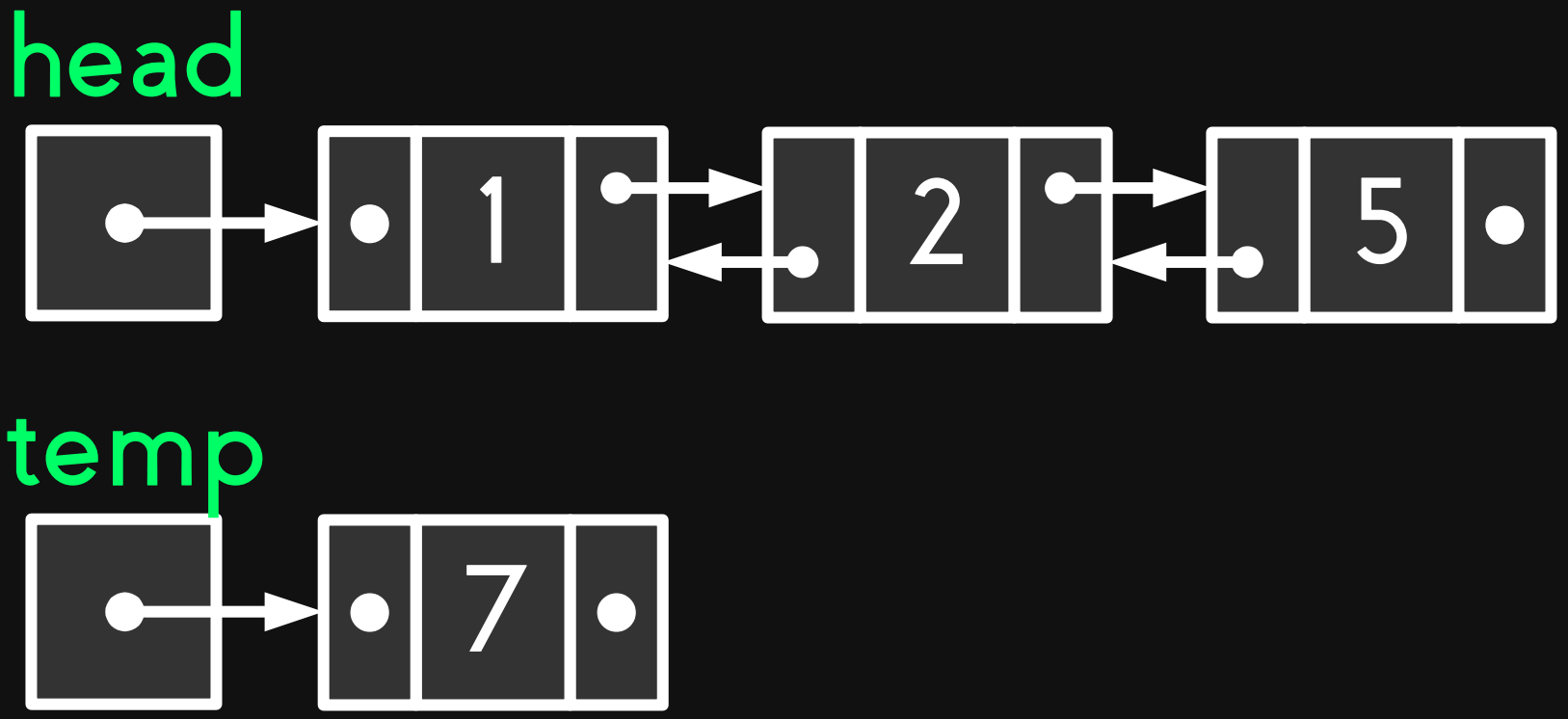
BACKWARD

Insert at Head



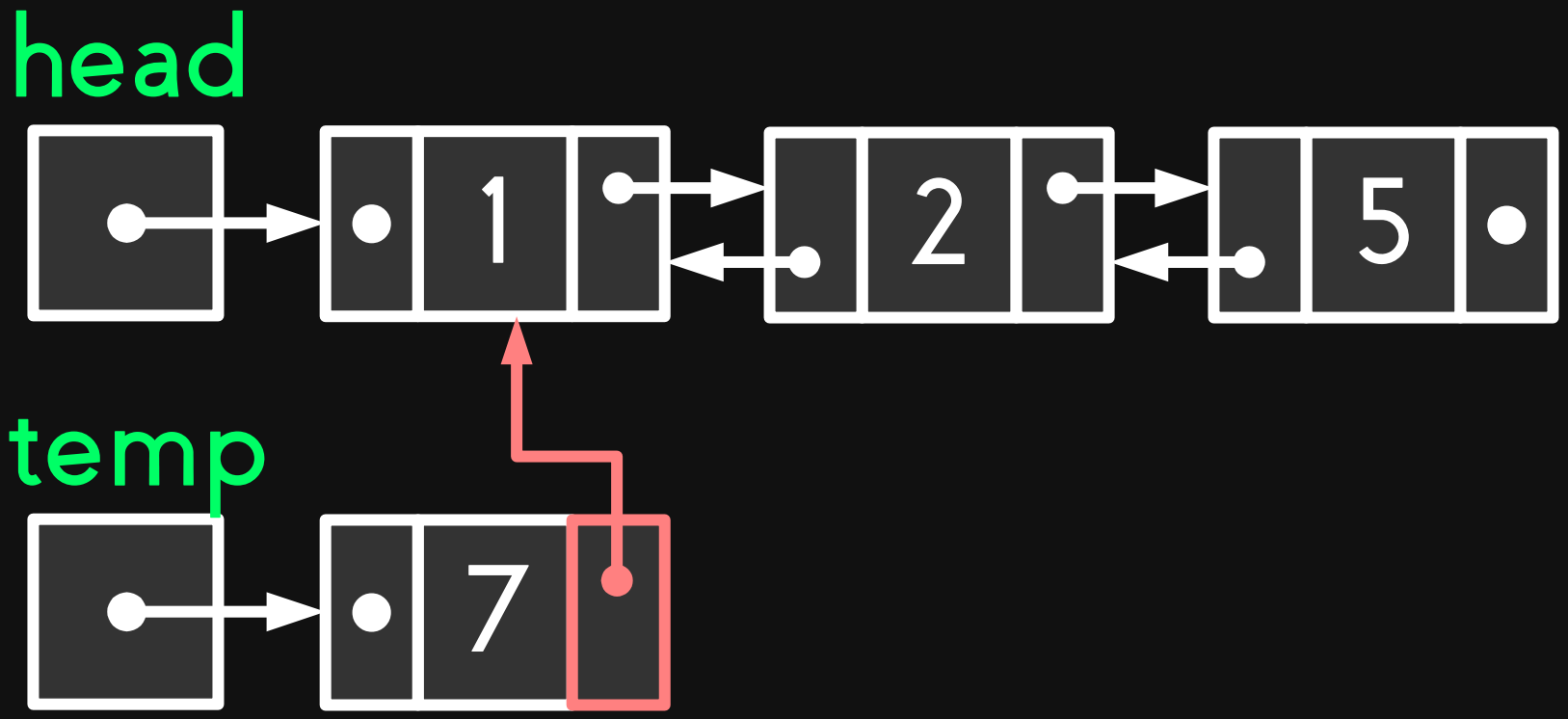
insert 7 at head

Insert at Head



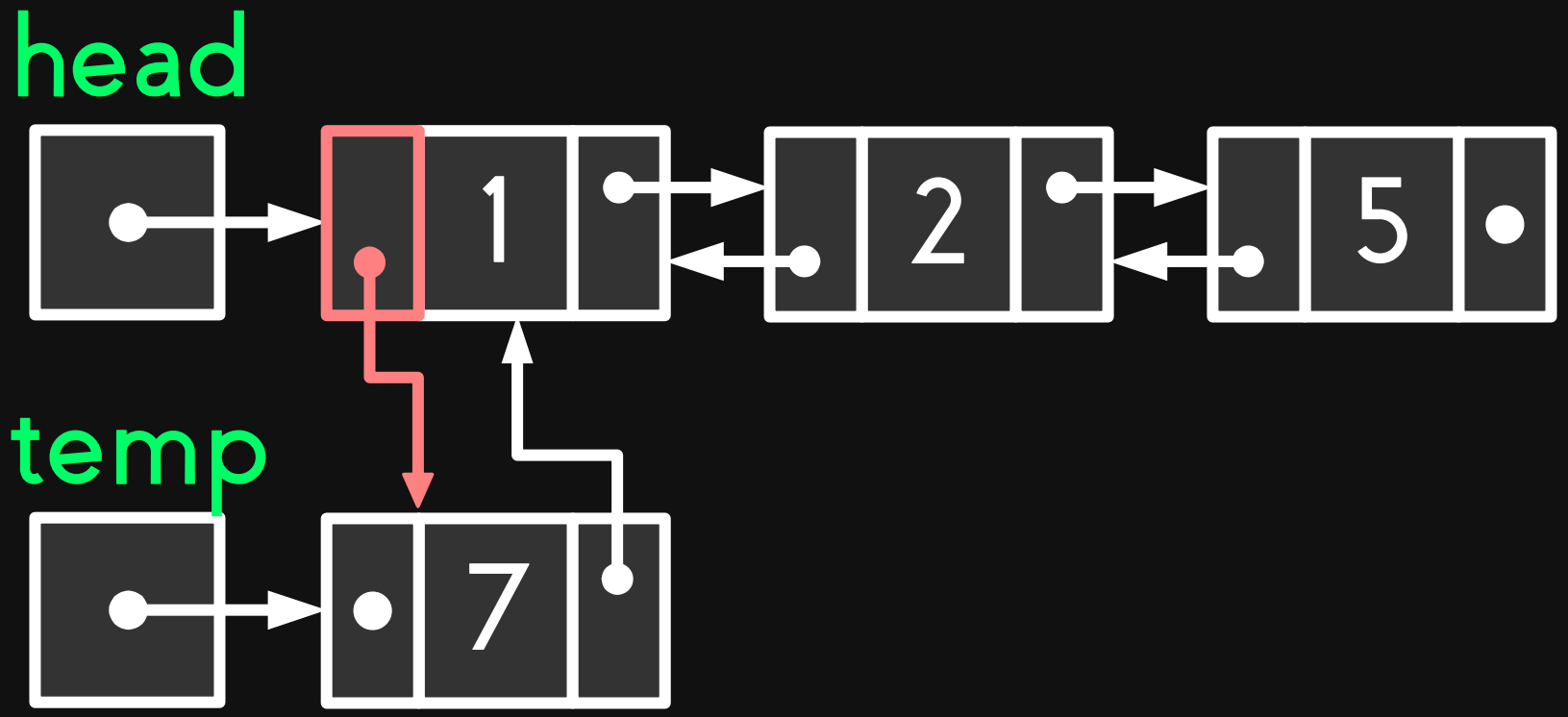
`malloc()` a node to a pointer (`temp`)
make sure the pointers are NULL

Insert at Head



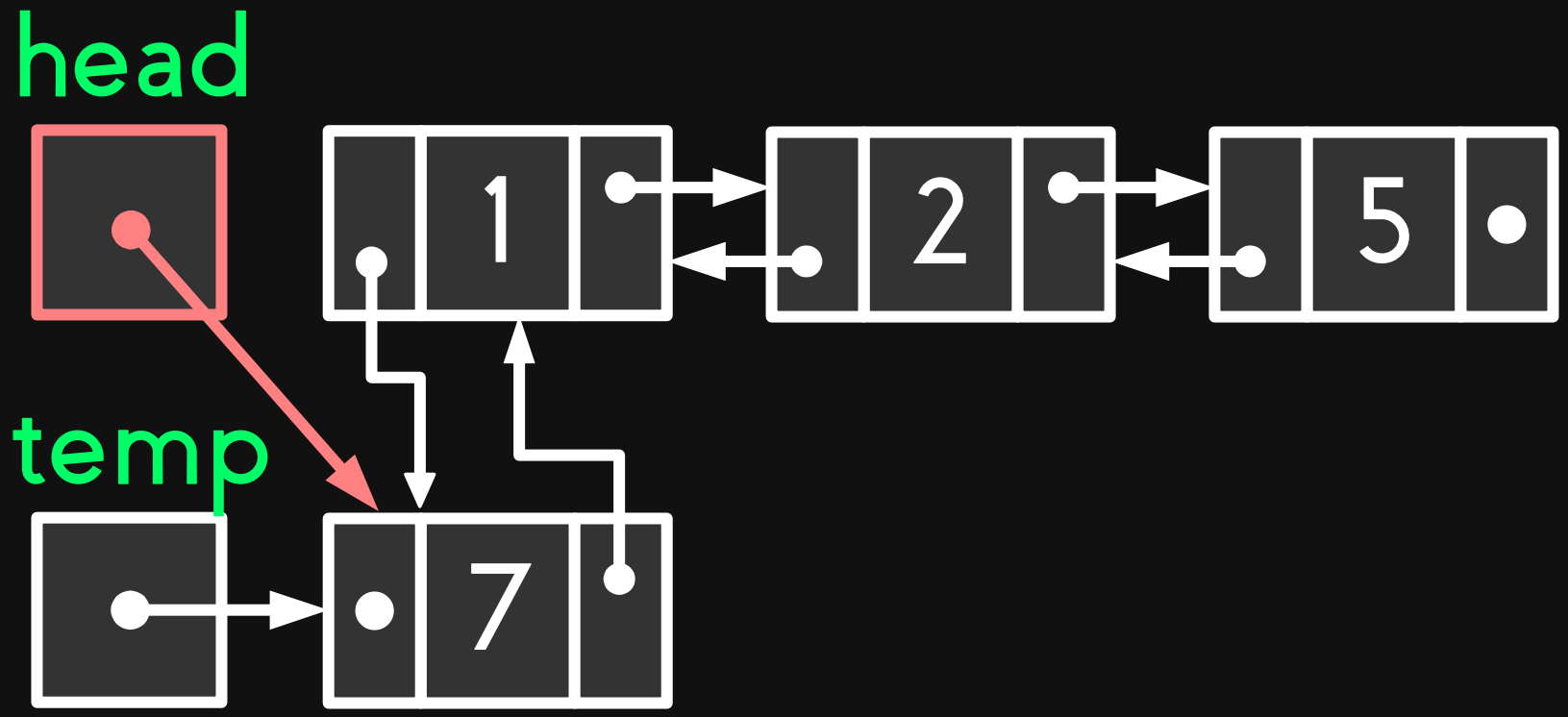
point the **next** pointer of the node being pointed by temp **to the current head node**

Insert at Head



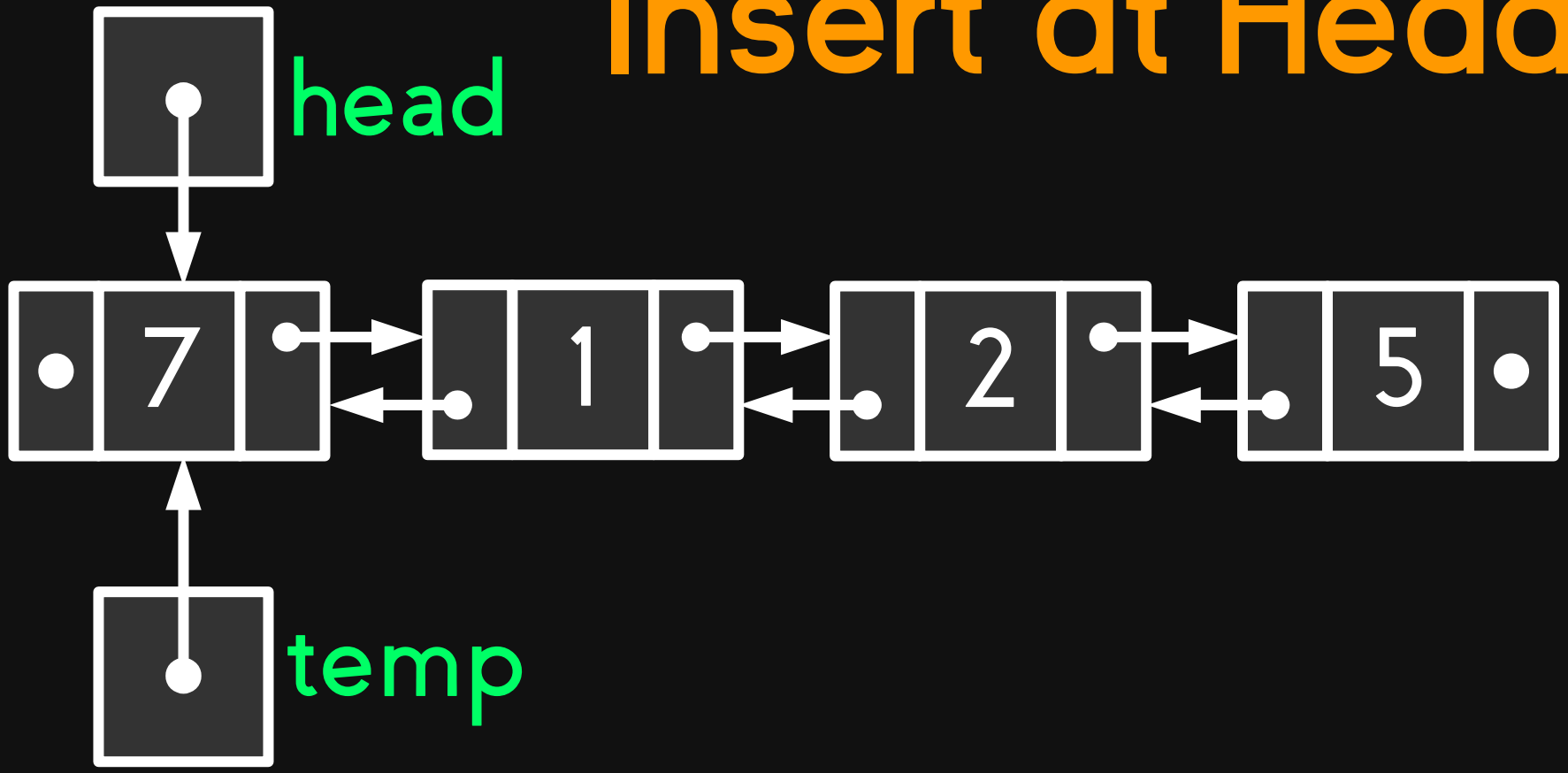
point the **prev** pointer of the current head node **to the new node**

Insert at Head



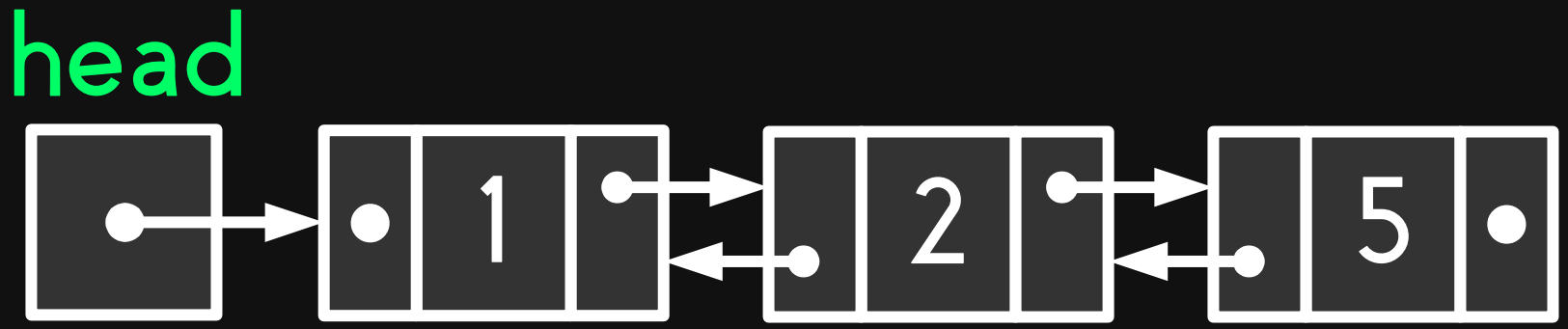
point the head pointer to the new node

Insert at Head



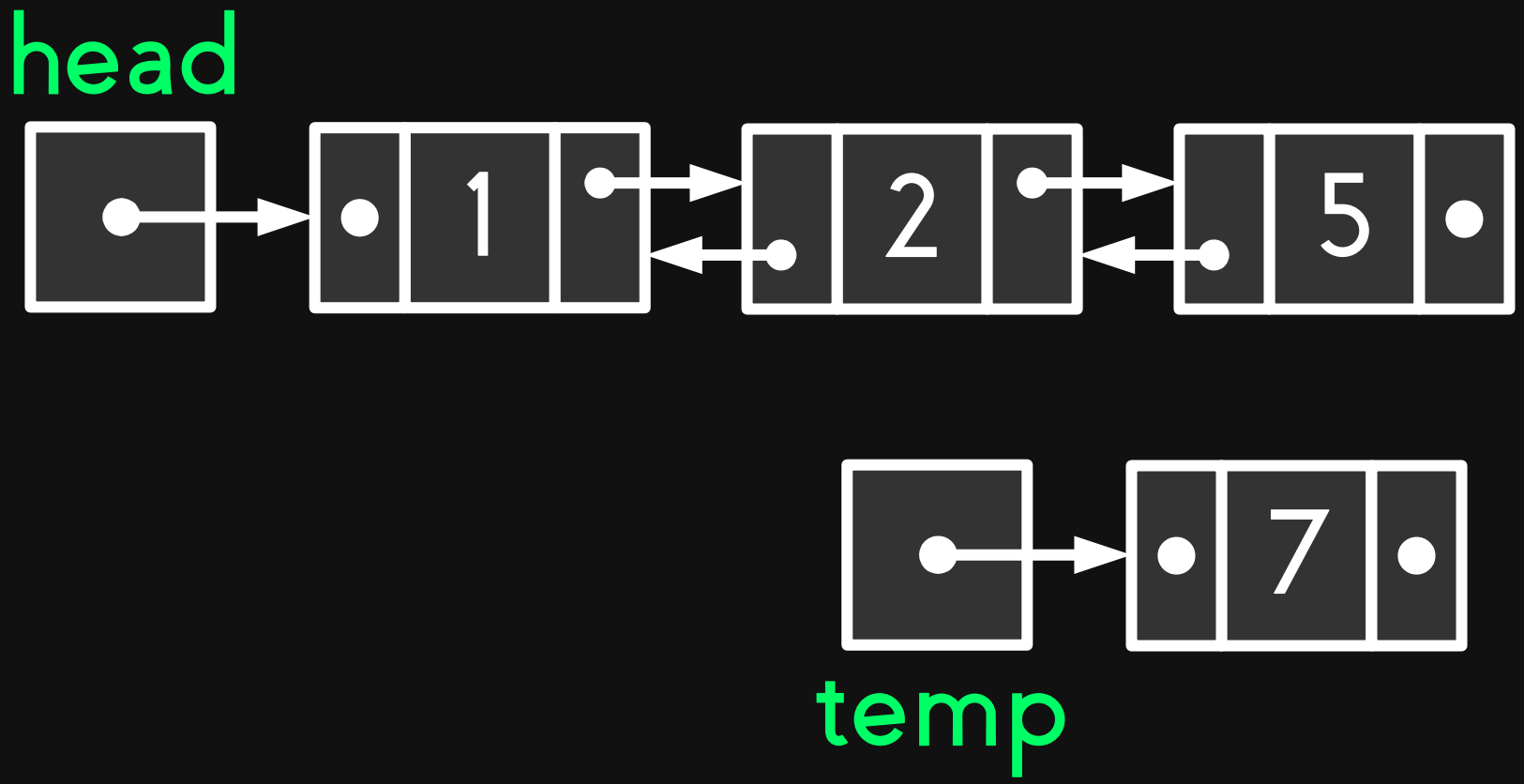
Rearrangement of the nodes.

Insert at Middle



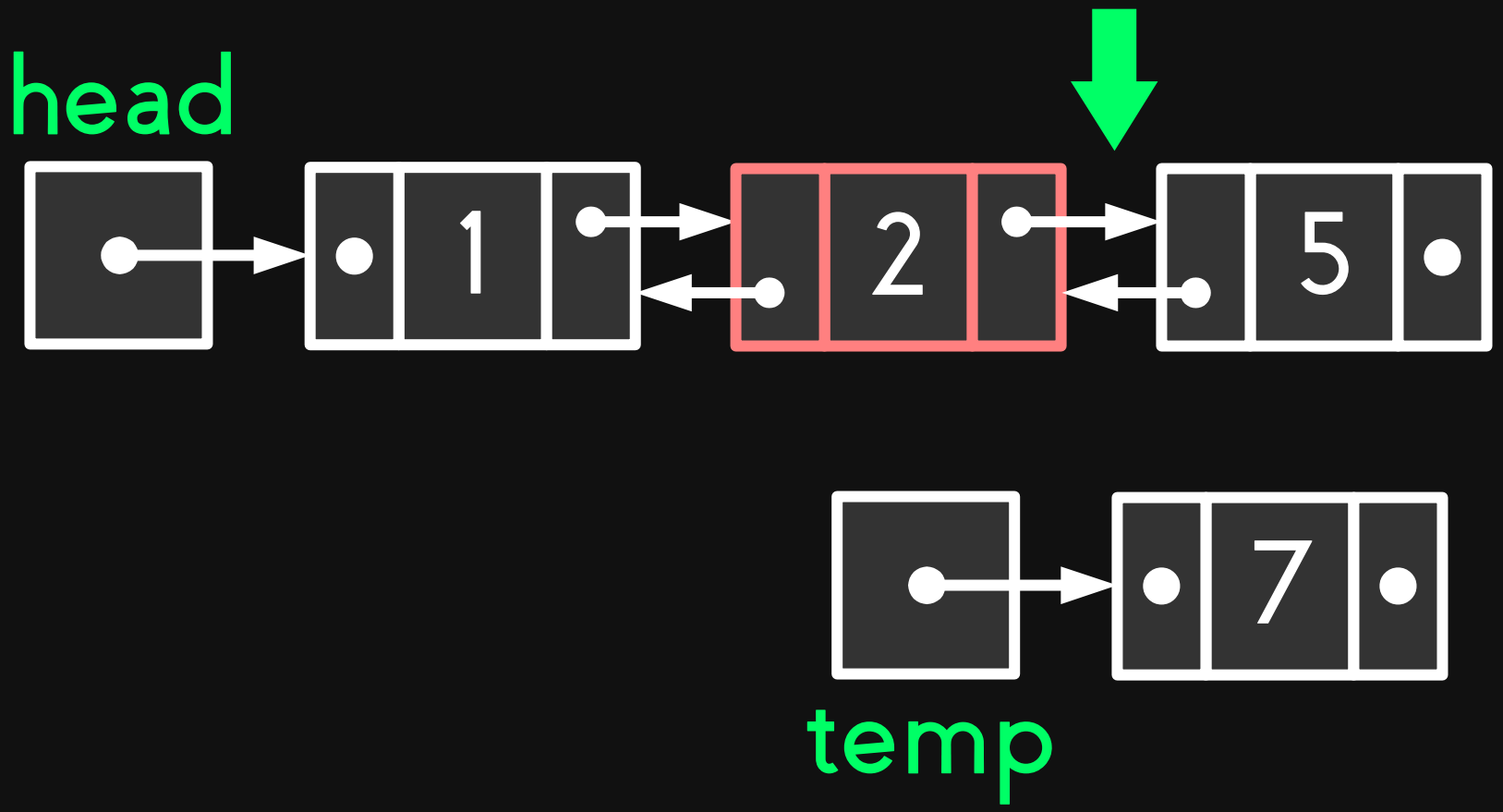
insert 7 in
between 2 and 5

Insert at Middle



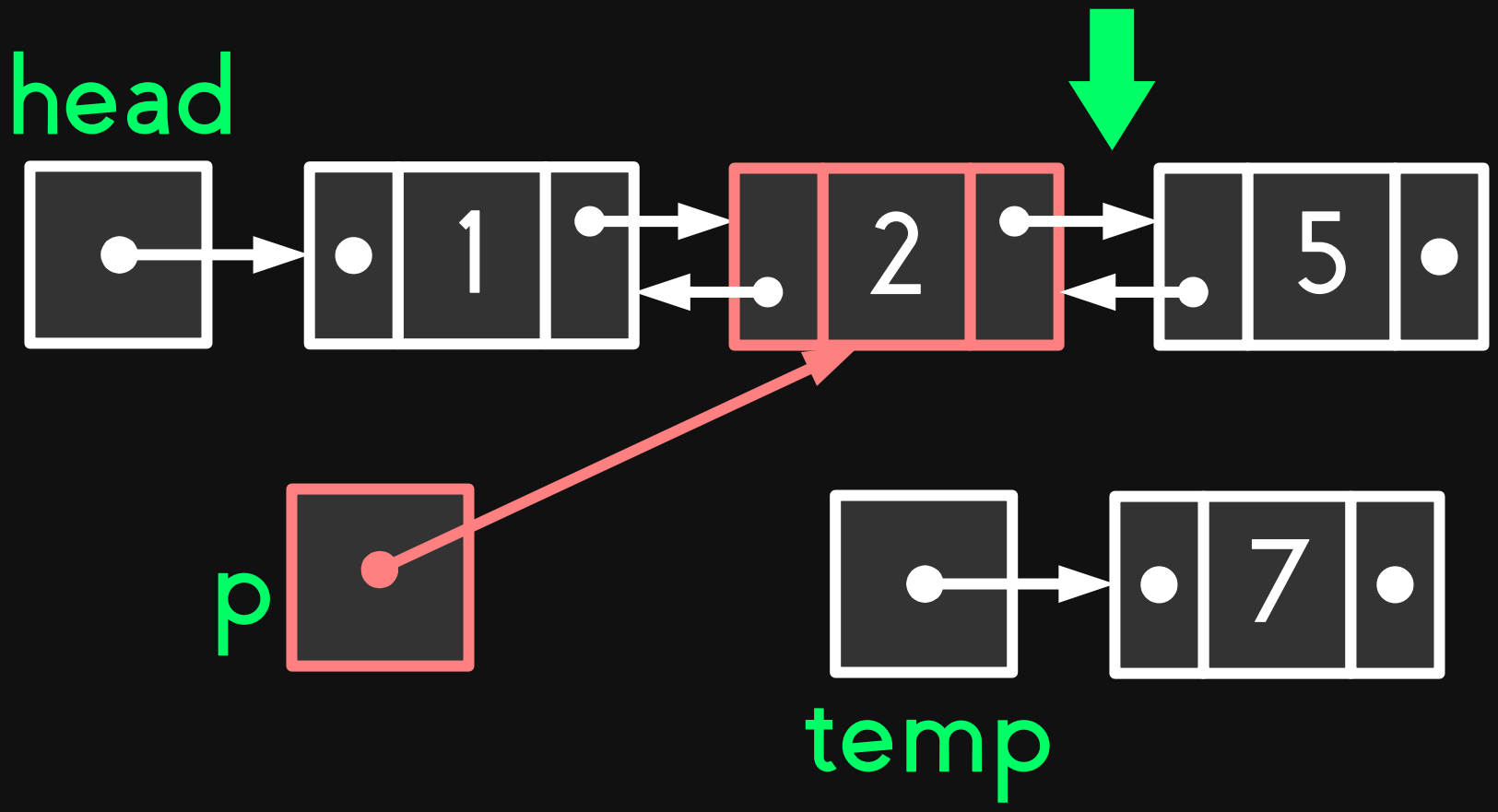
`malloc()` a node to a pointer (`temp`)
make sure the pointers are NULL

Insert at Middle



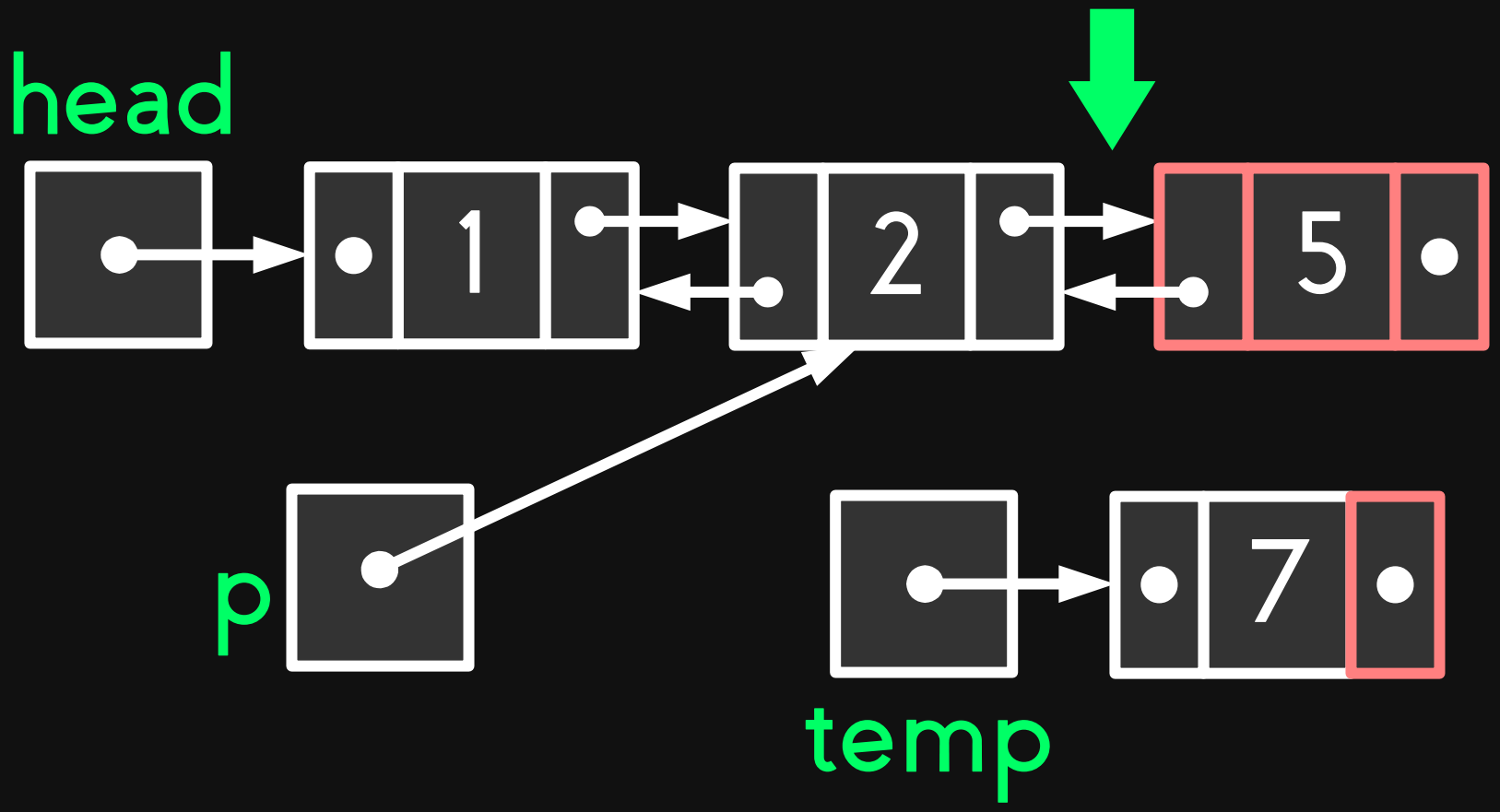
Find the node before the position you want the new node to be inserted.

Insert at Middle



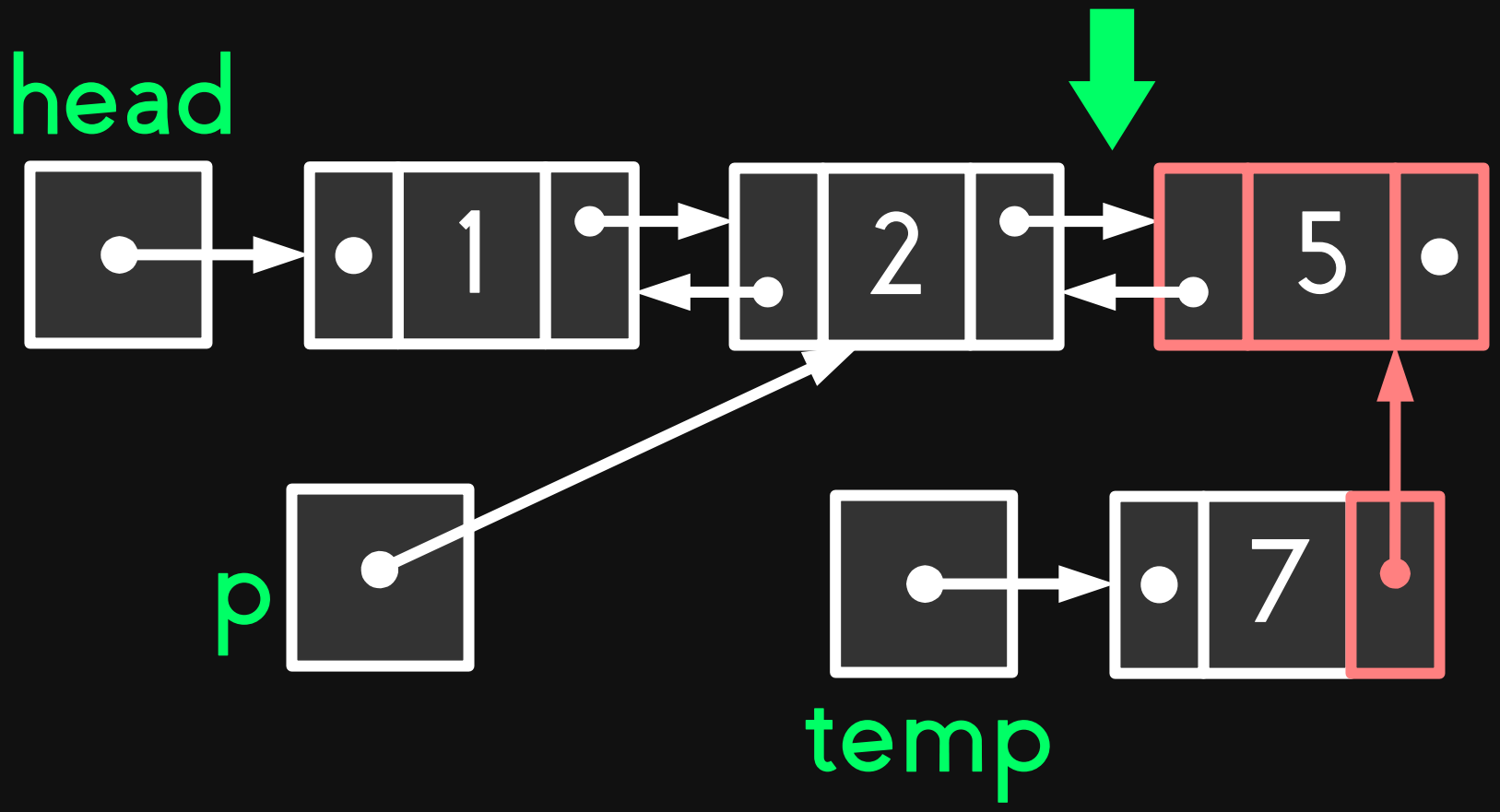
Let a pointer (p) point to that node.

Insert at Middle



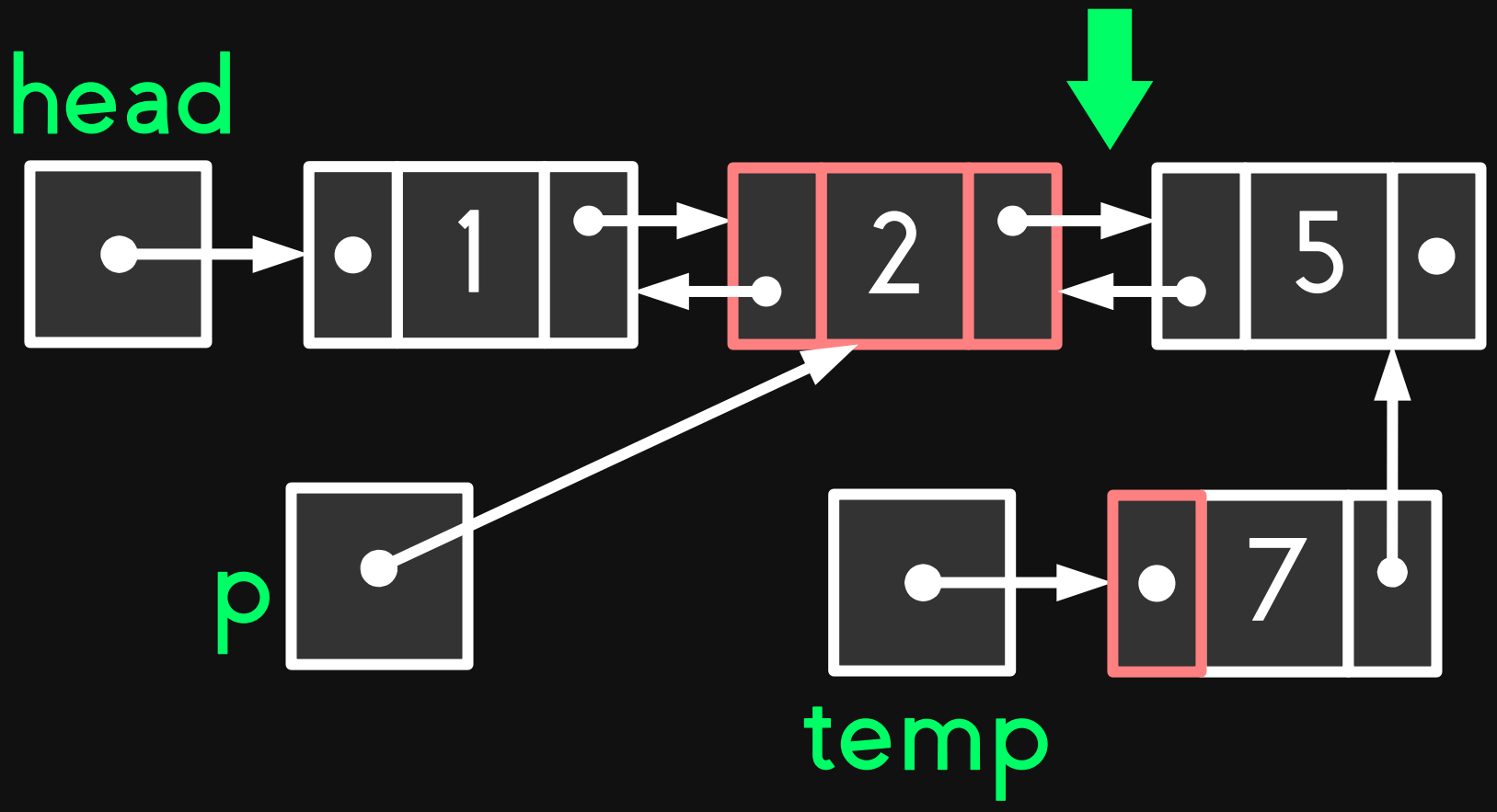
point the next of the new node to the node being pointed by the next pointer of the node being pointed by p

Insert at Middle



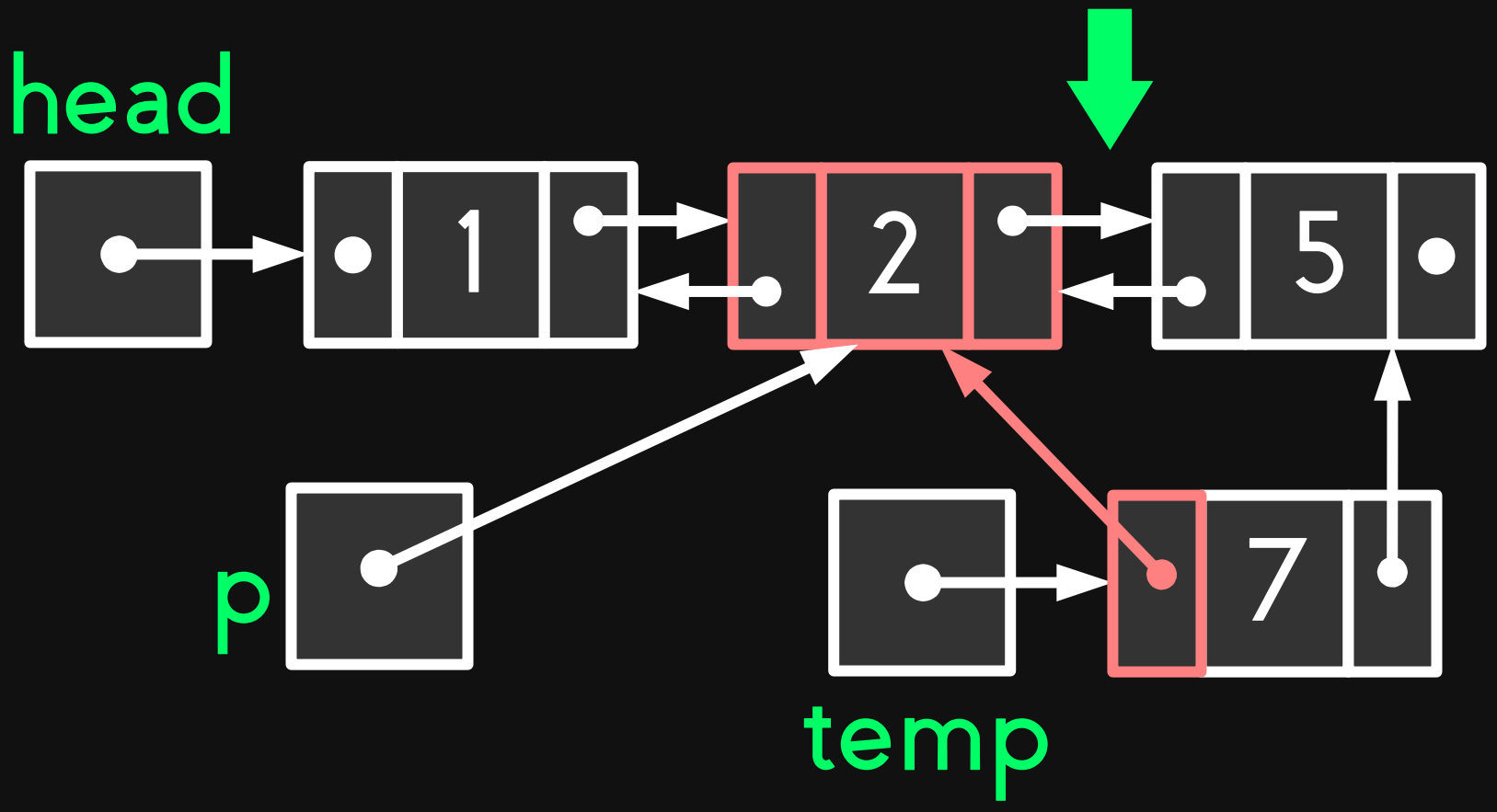
point the next of the new node to the node being pointed by the next pointer of the node being pointed by p

Insert at Middle



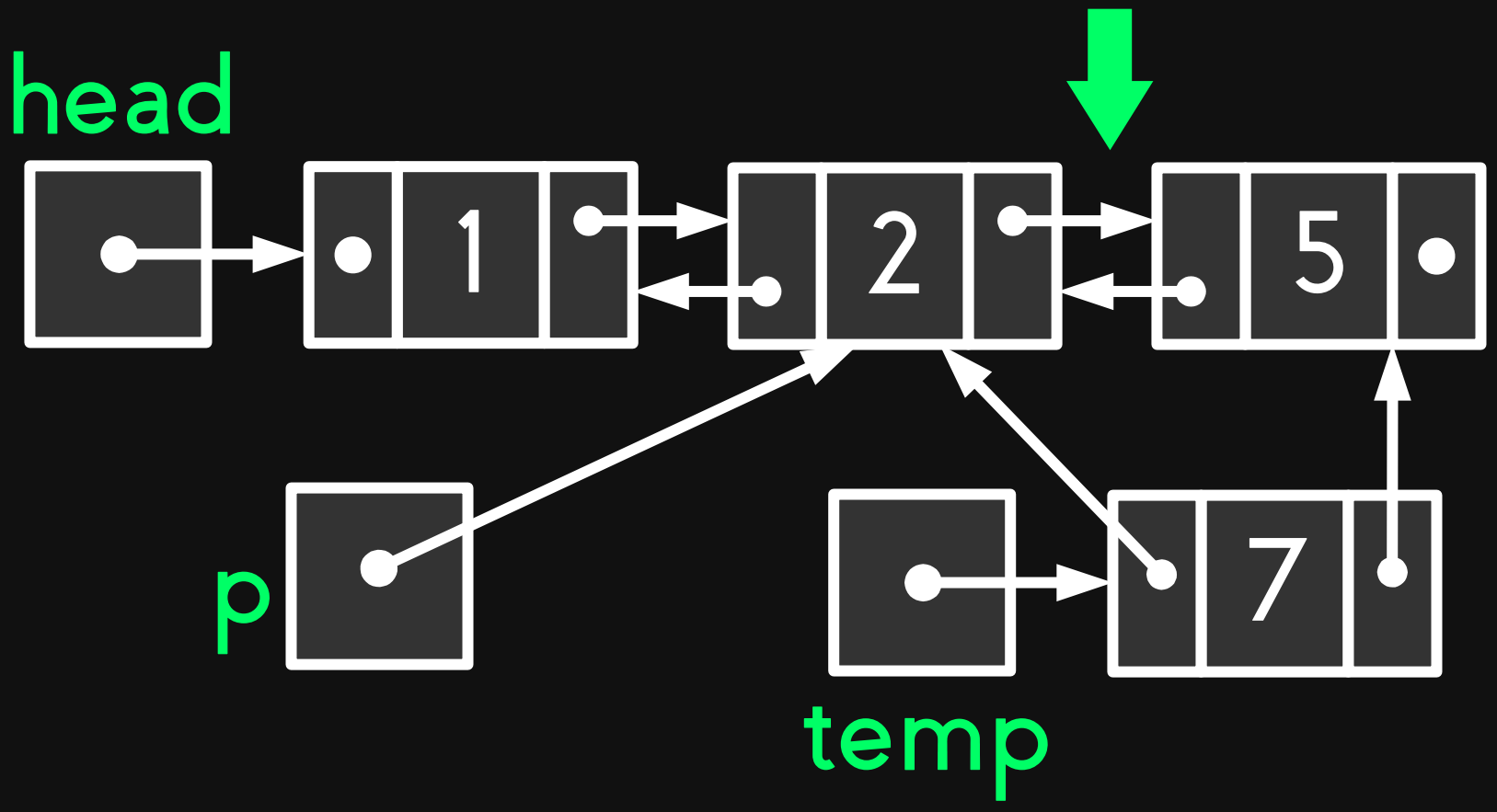
point the **prev of the new node** to the
node being pointed by p

Insert at Middle



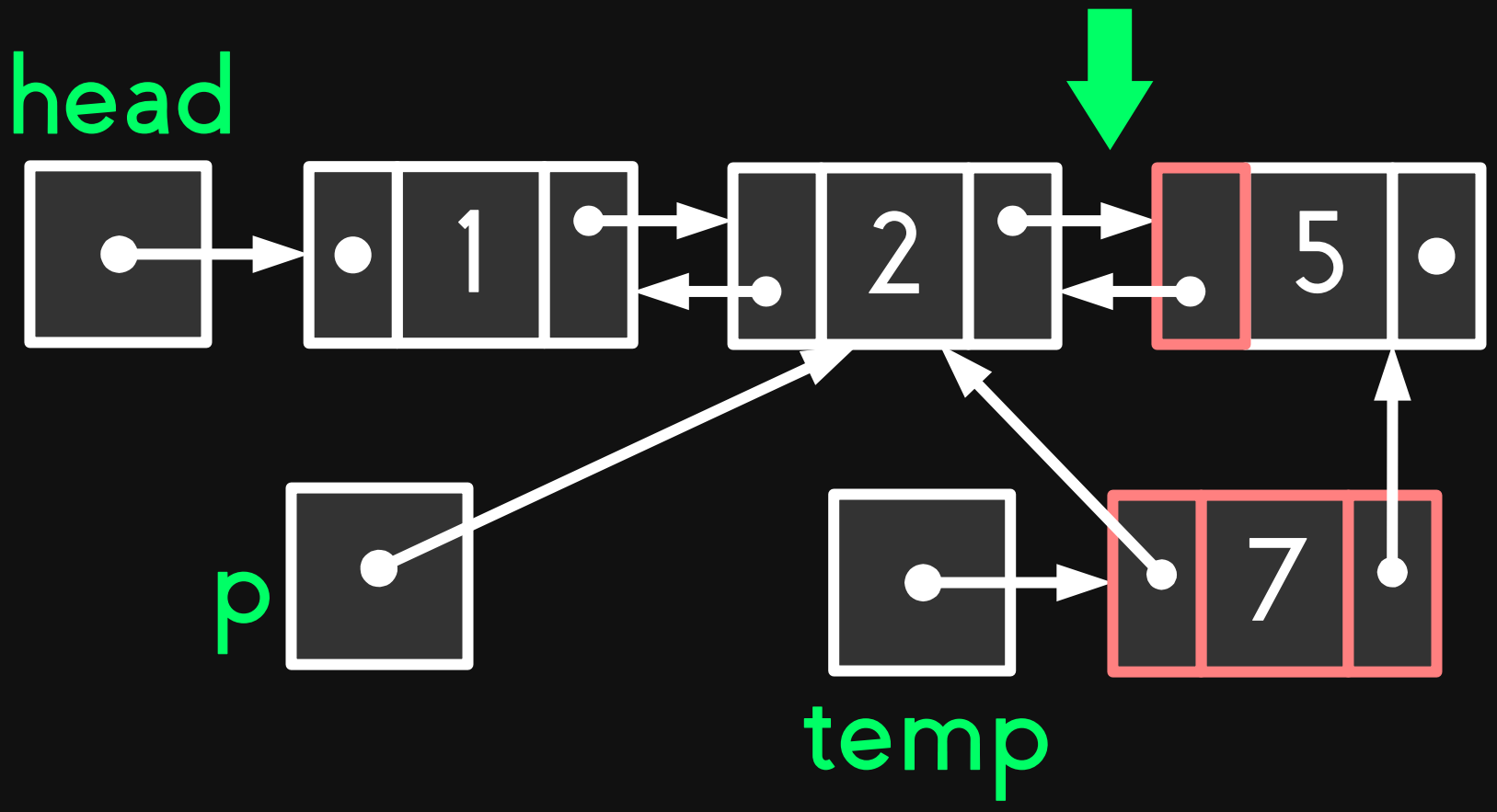
point the **prev of the new node** to the
node being pointed by p

Insert at Middle



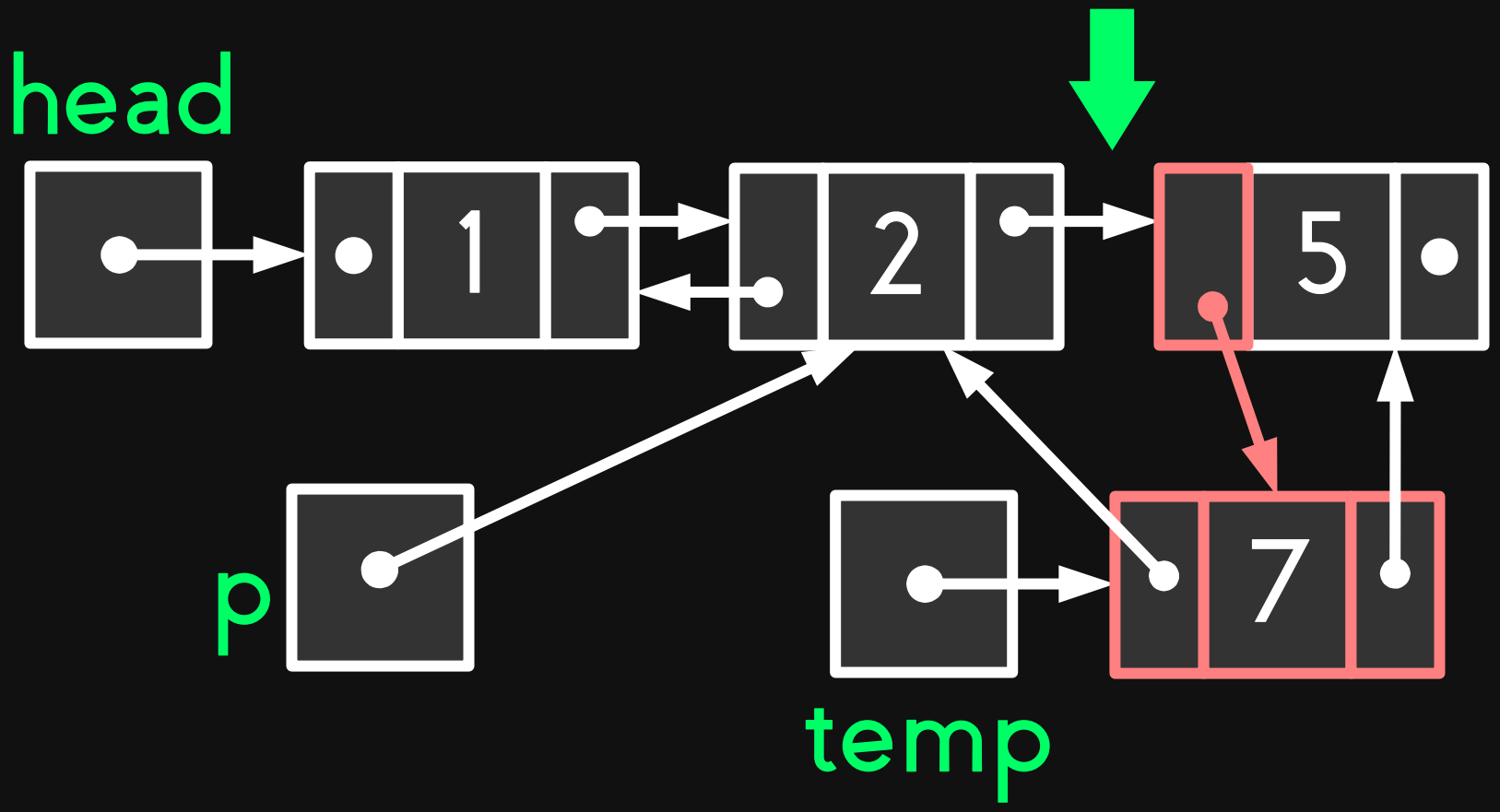
What should be changed next?

Insert at Middle



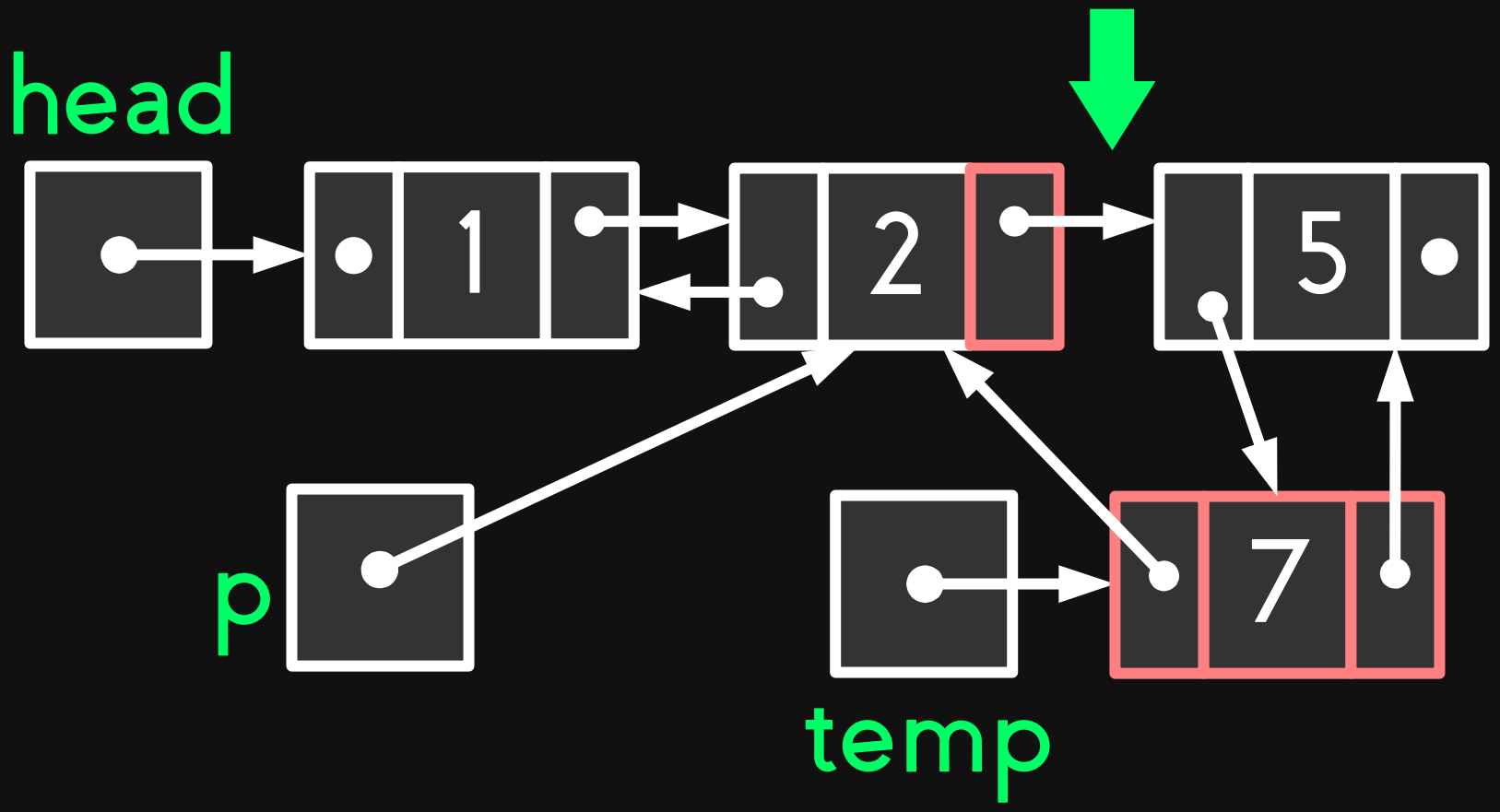
point the **prev of the node being pointed by the next of the node being pointed by p** to the **new node**

Insert at Middle



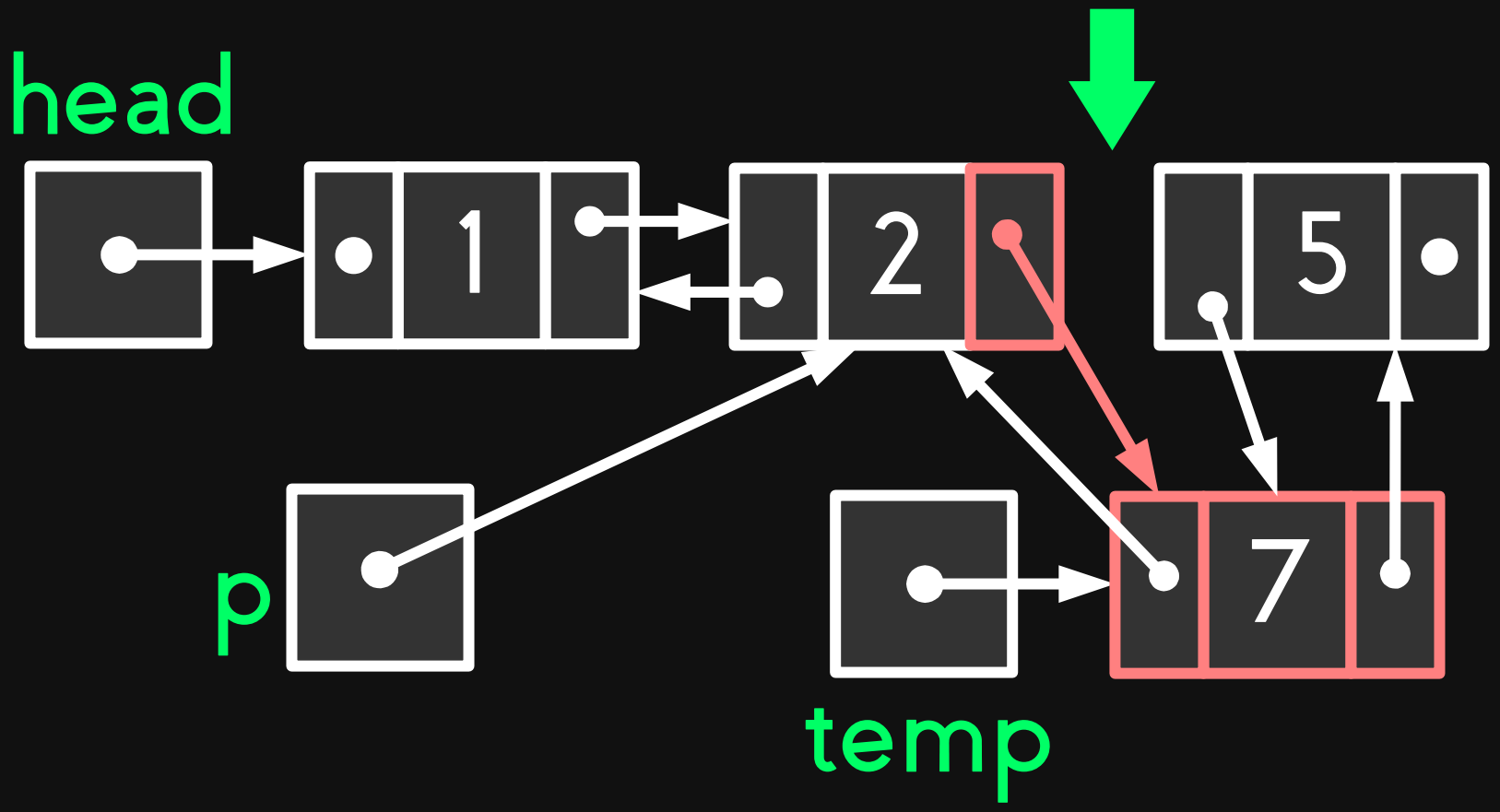
point the **prev of the node being pointed by the next of the node being pointed by p** to the **new node**

Insert at Middle



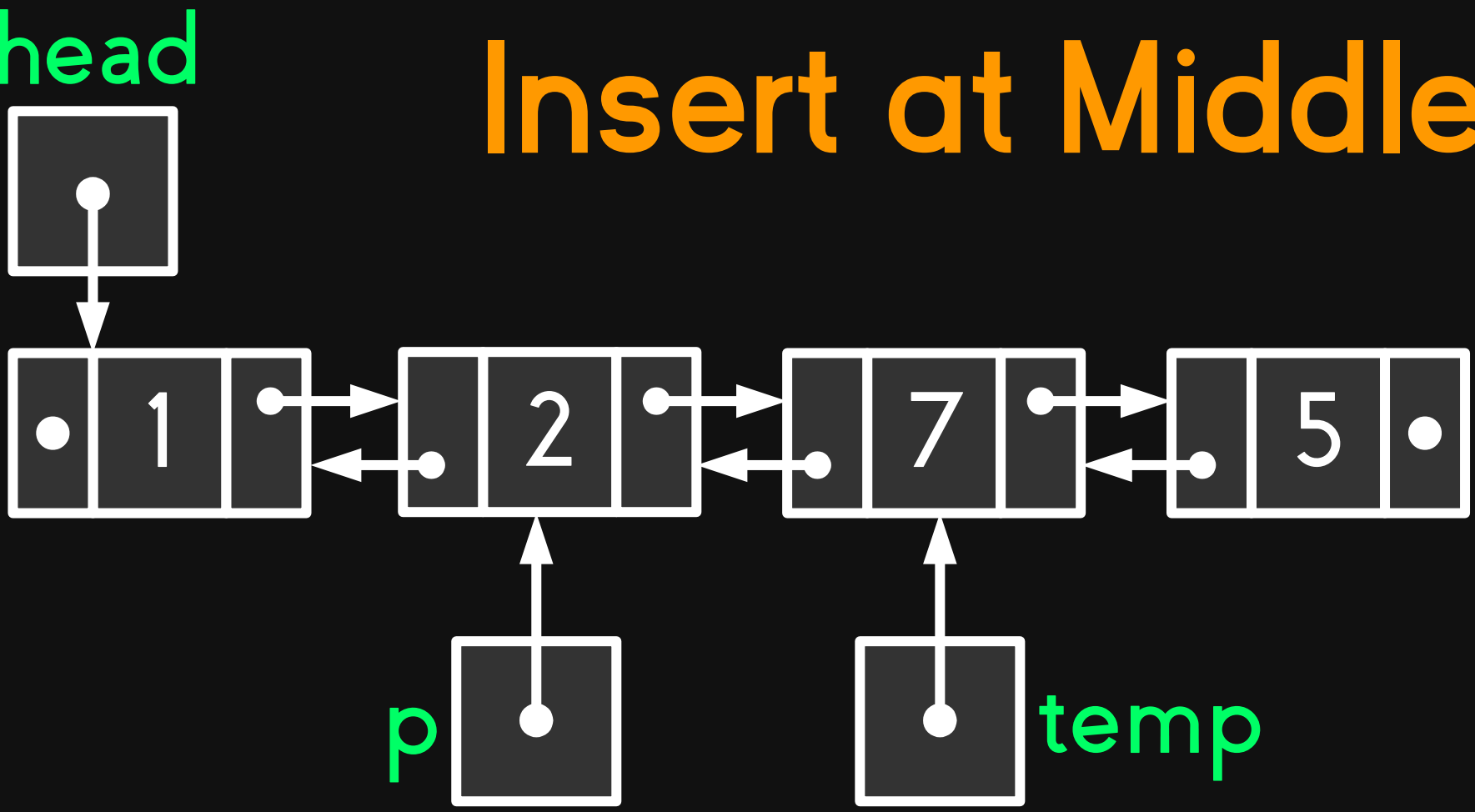
point the next of the node being pointed by p to the new node

Insert at Middle



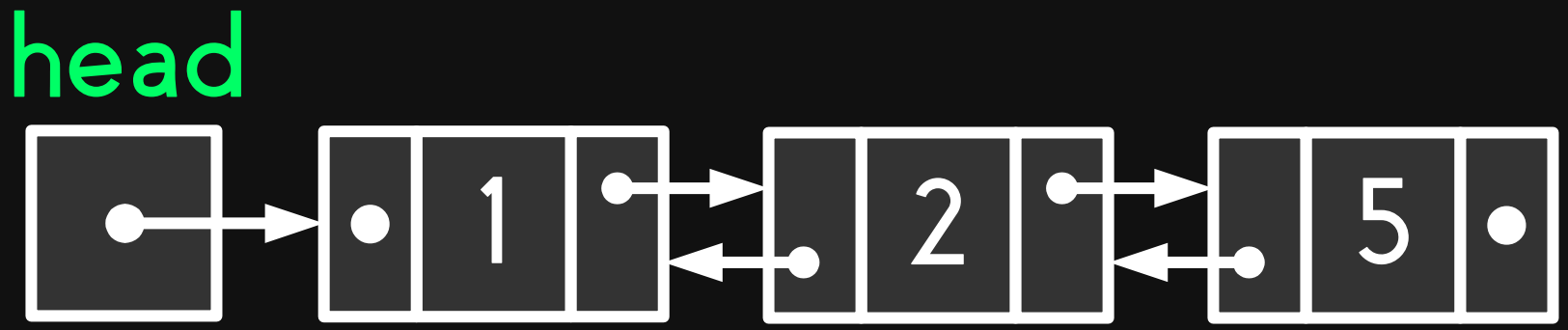
point the next of the node being pointed by p to the new node

Insert at Middle



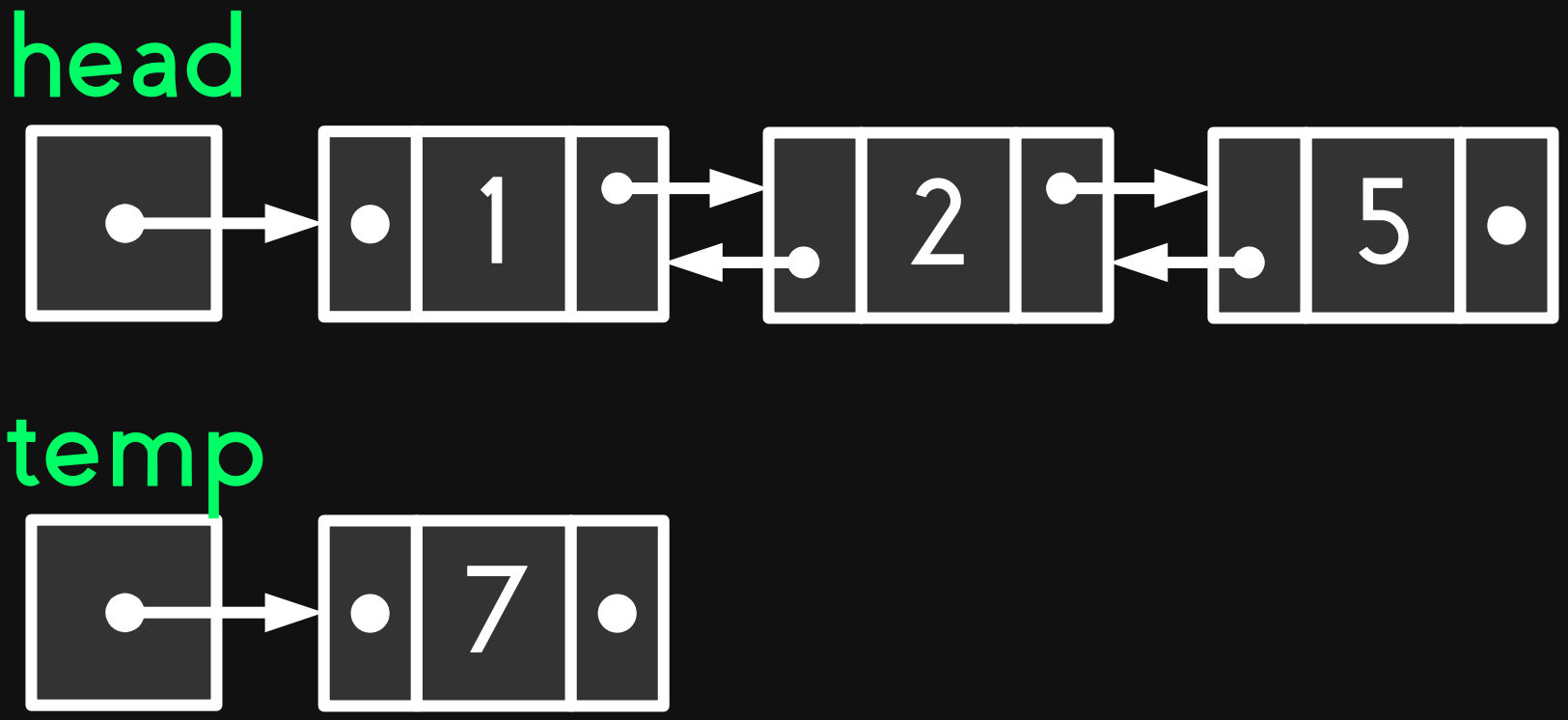
Rearrangement of the nodes.

Insert at Tail



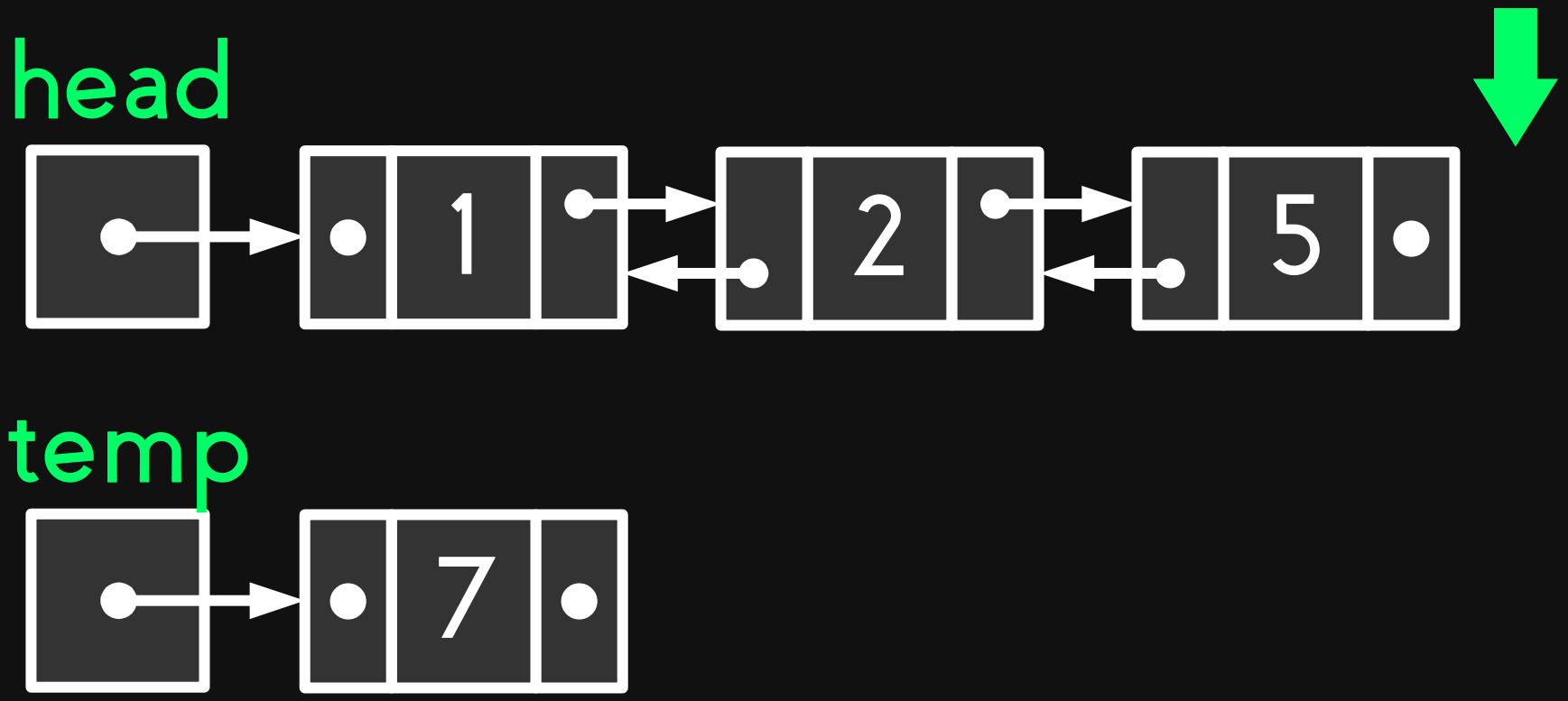
insert 7 at tail

Insert at Tail



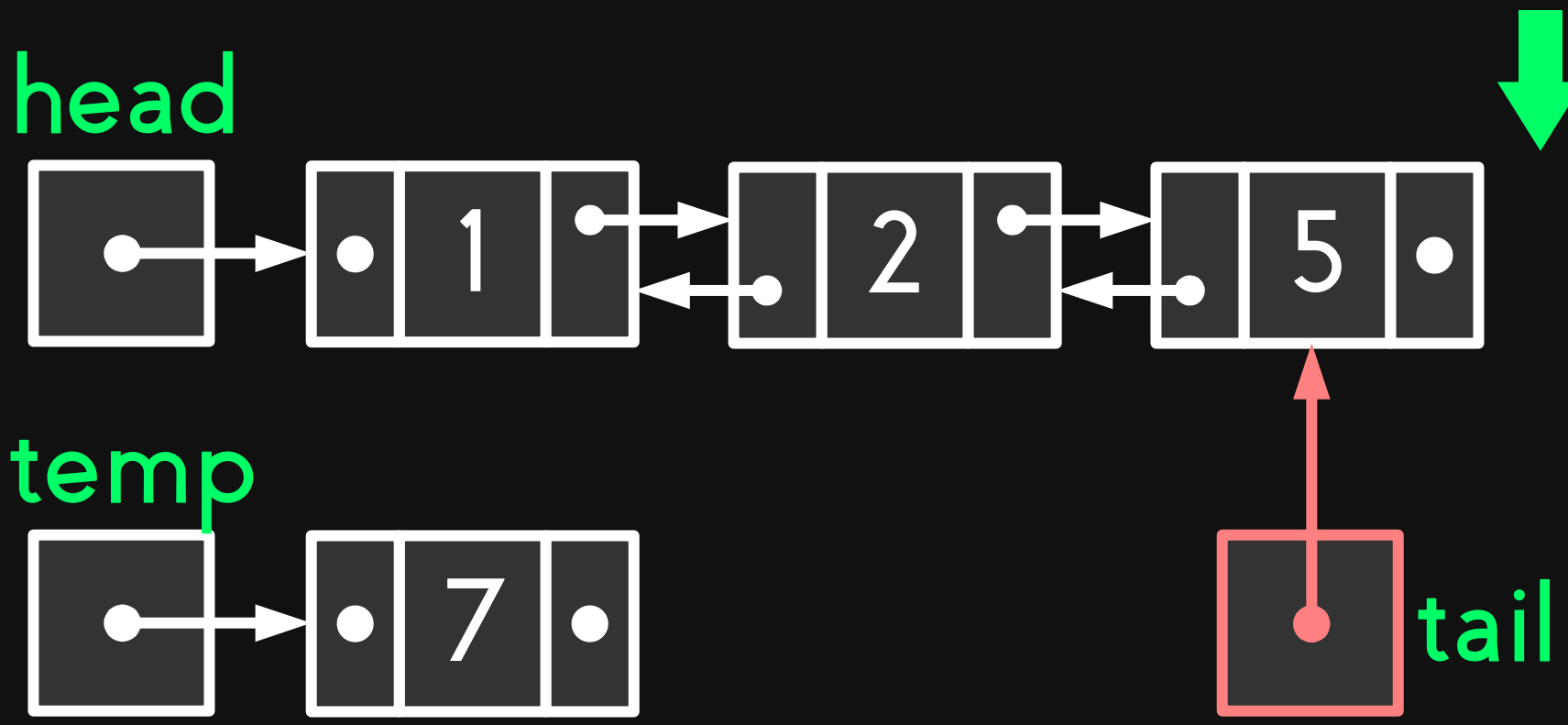
`malloc()` a node to a pointer (`temp`)
make sure the pointers are NULL

Insert at Tail



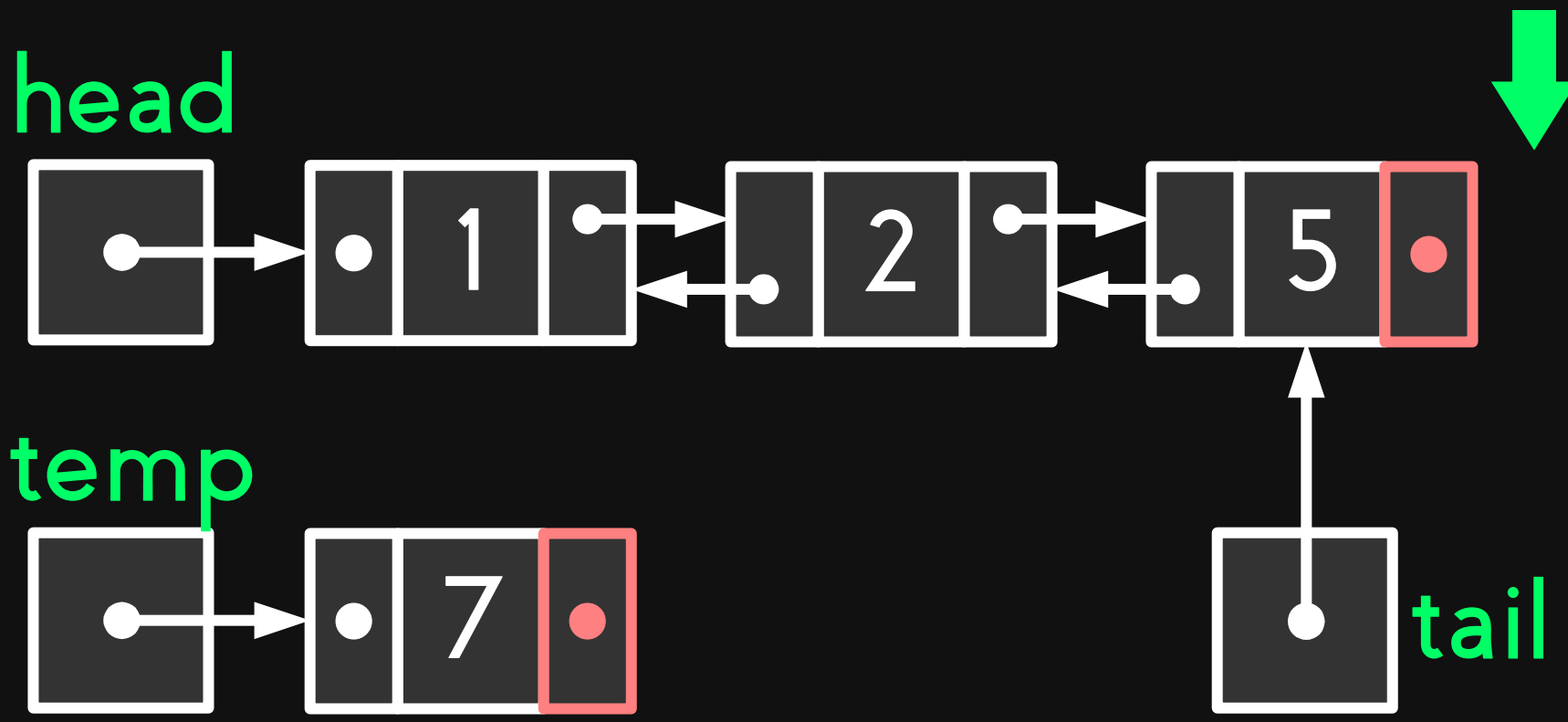
Find the node before the position you want the new node to be inserted.

Insert at Tail



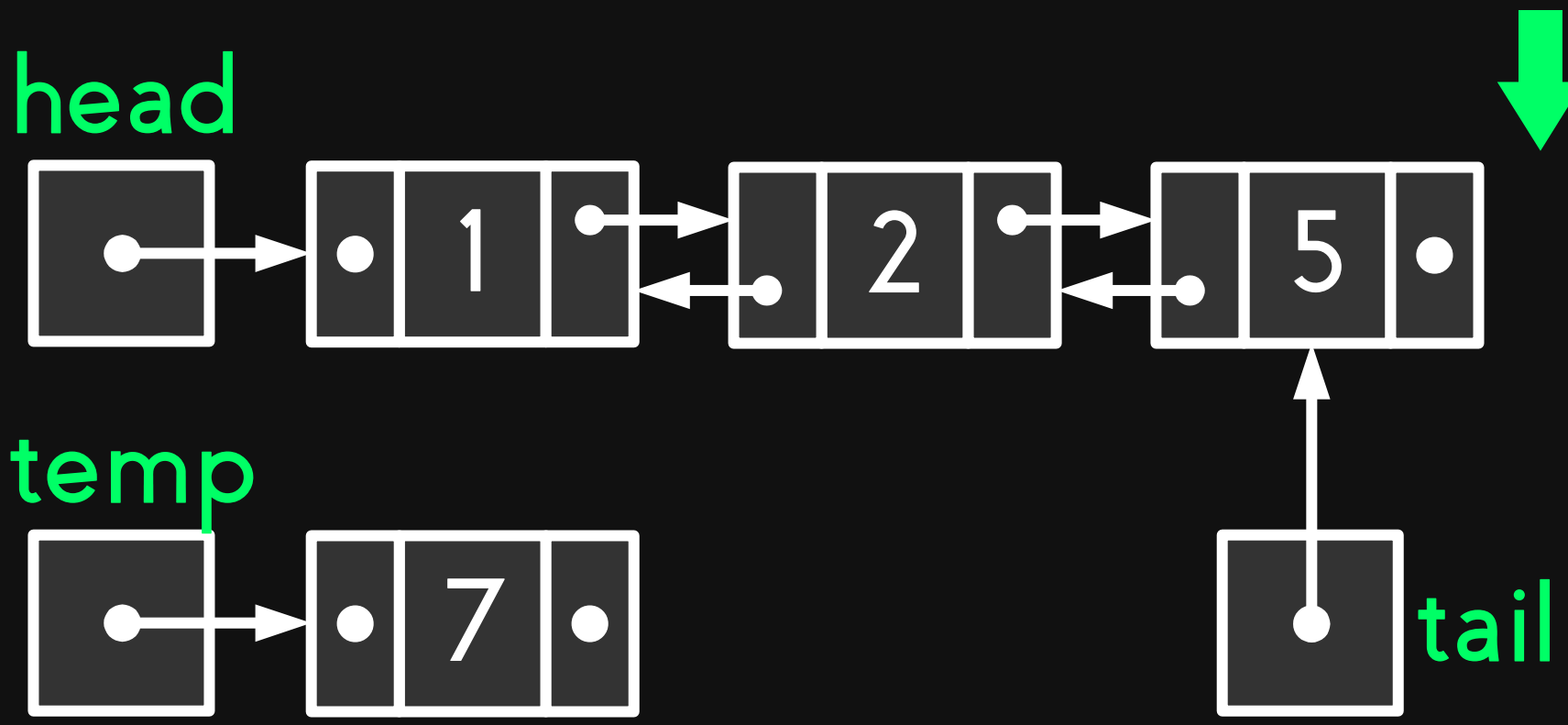
Let a pointer (**tail**) point to the tail node.

Insert at Tail



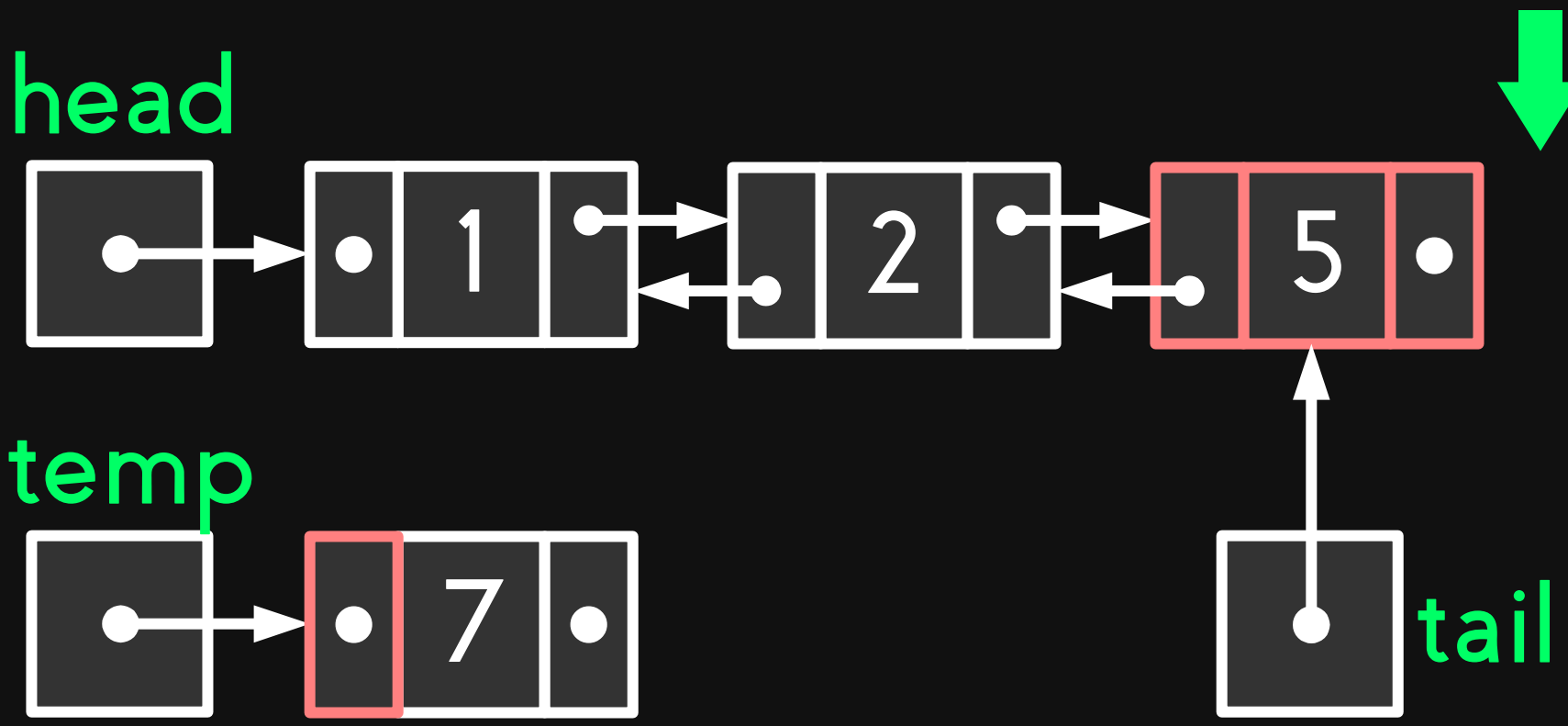
point the next of the new node to the next of the tail node.

Insert at Tail



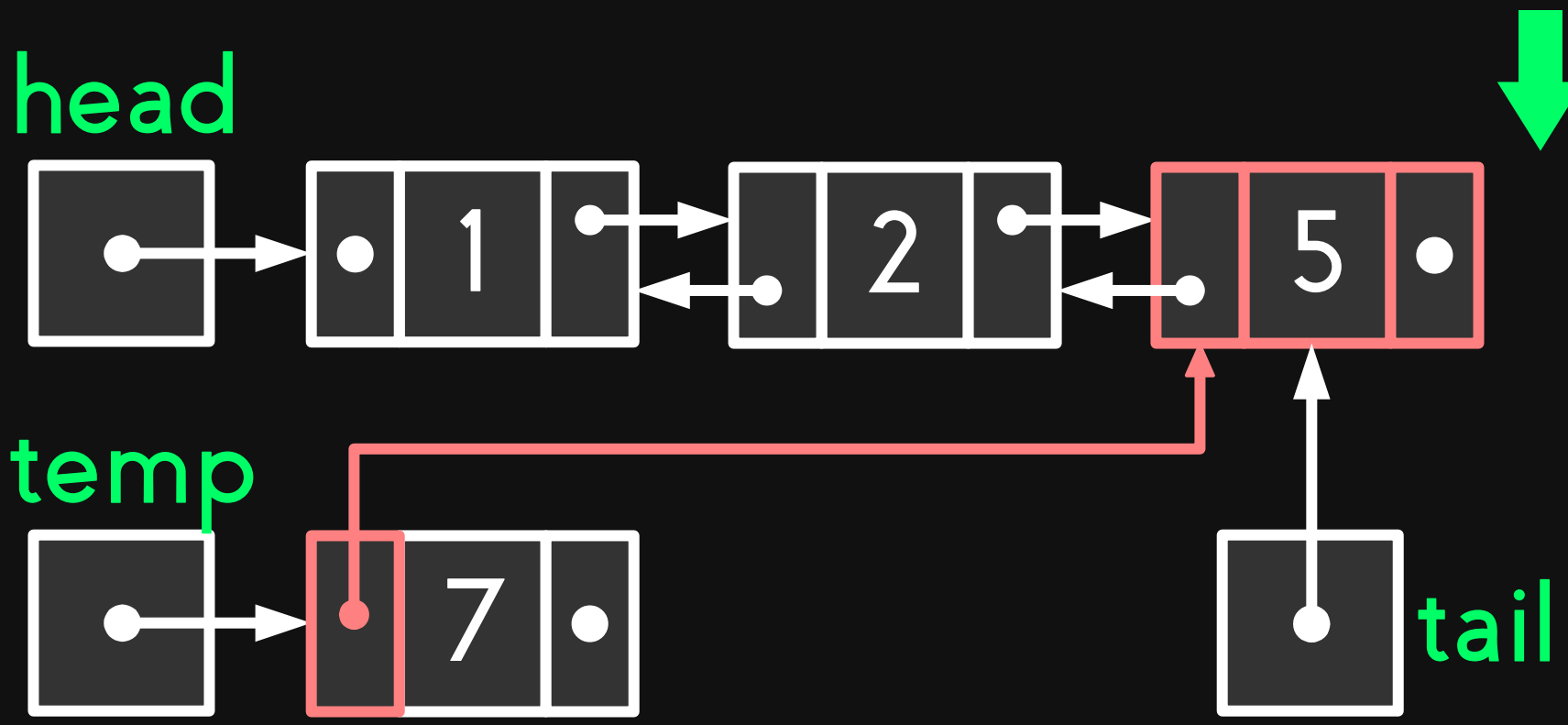
(it will have the same diagram!)

Insert at Tail



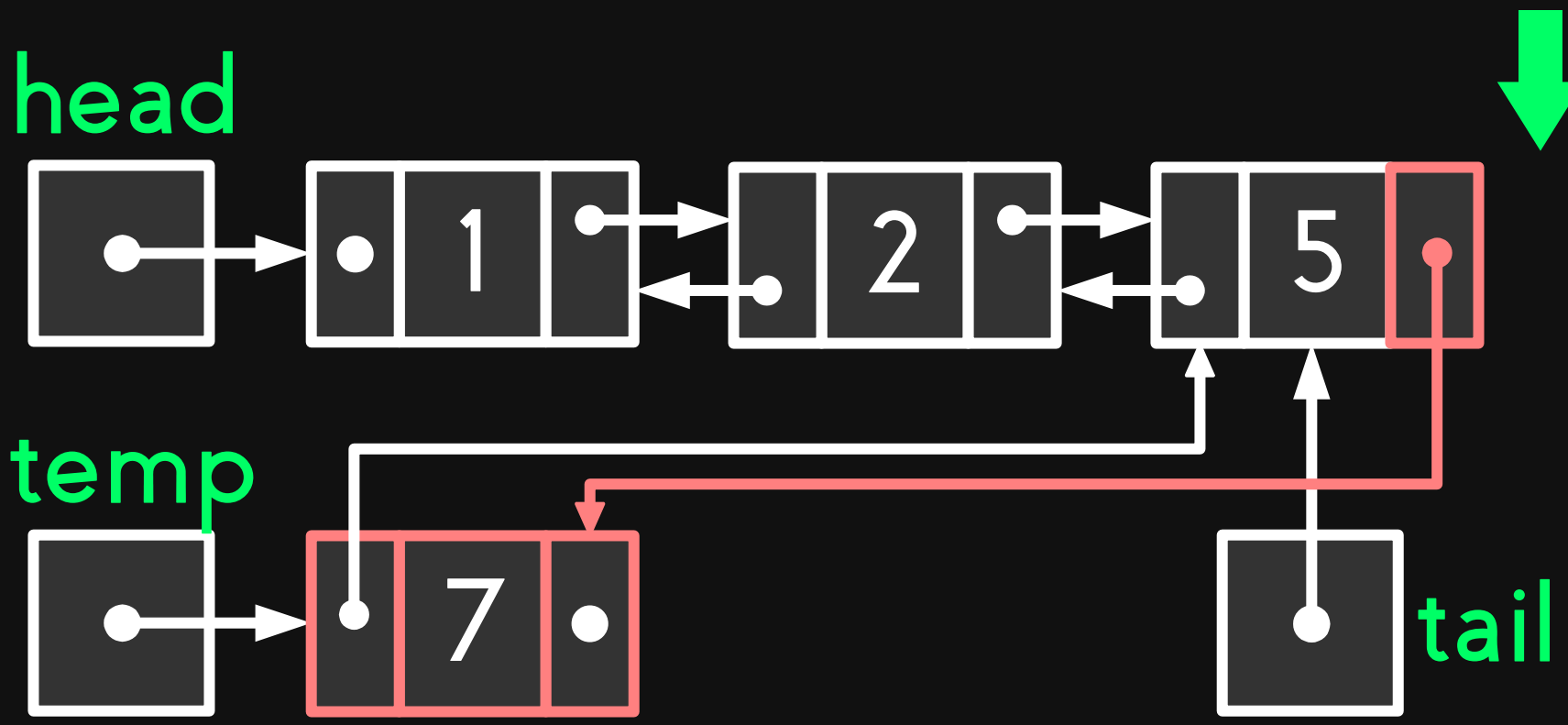
point the prev of the new node to the tail node.

Insert at Tail



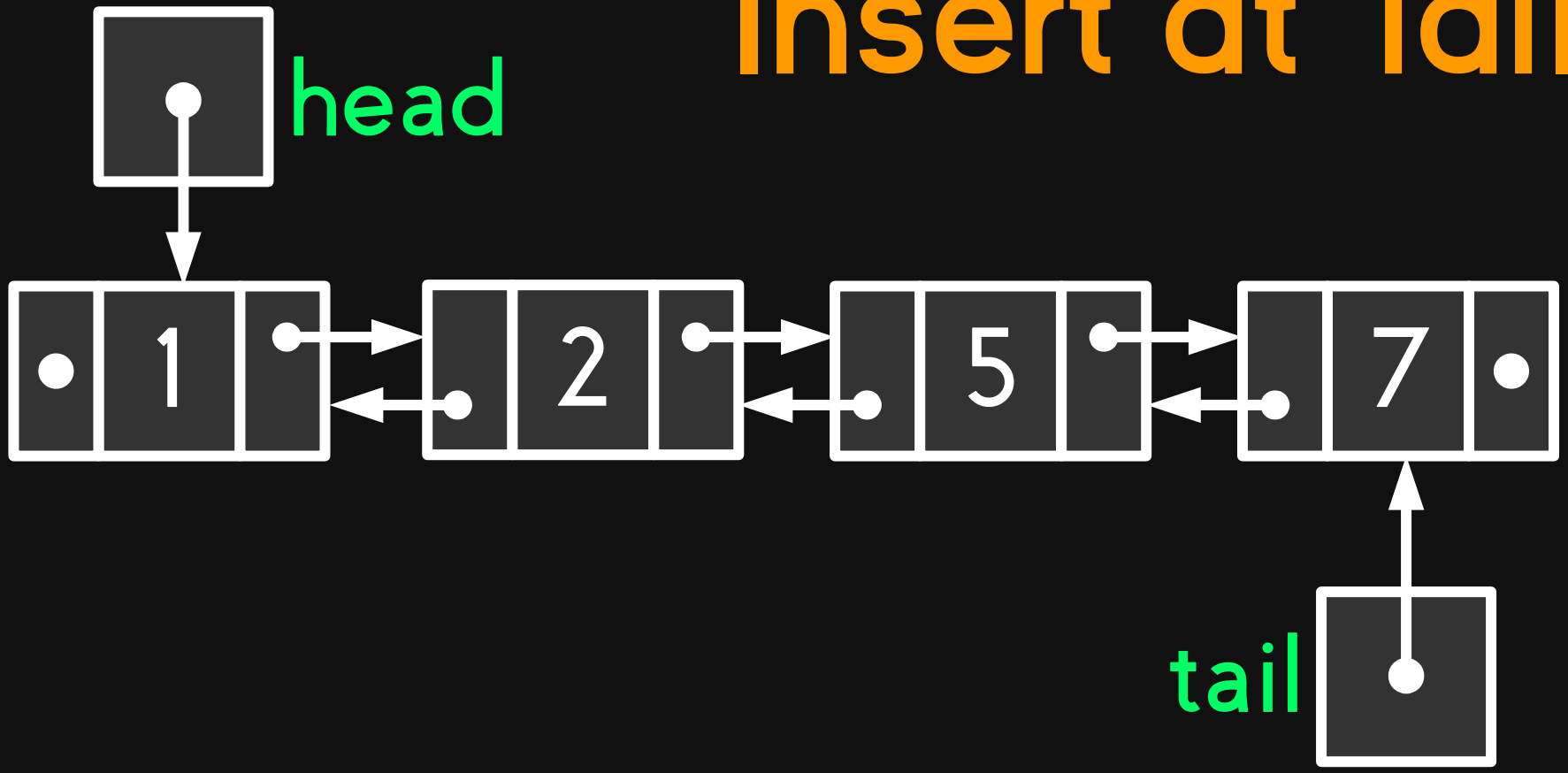
point the prev of the new node to the tail node.

Insert at Tail



point the **next of the tail node to the new node**.

Insert at Tail



Rearrangement of the nodes.

CMSC 21

FUNDAMENTALS OF PROGRAMMING

Kristine Bernadette P. Pelaez

Institute of Computer Science
University of the Philippines Los Baños