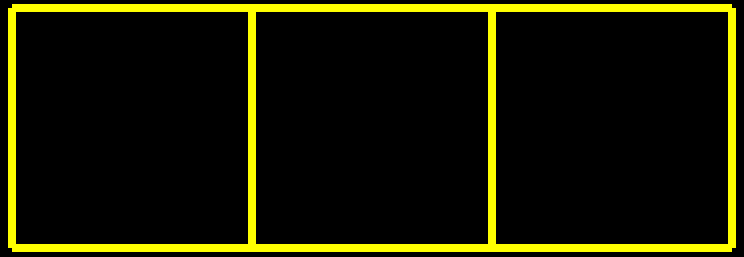
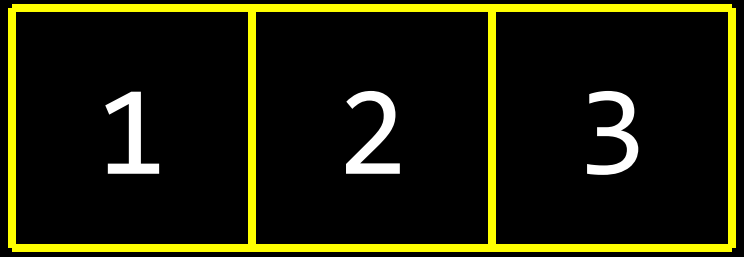
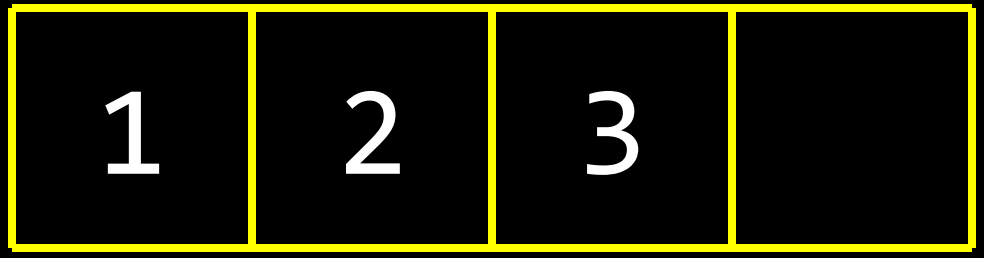
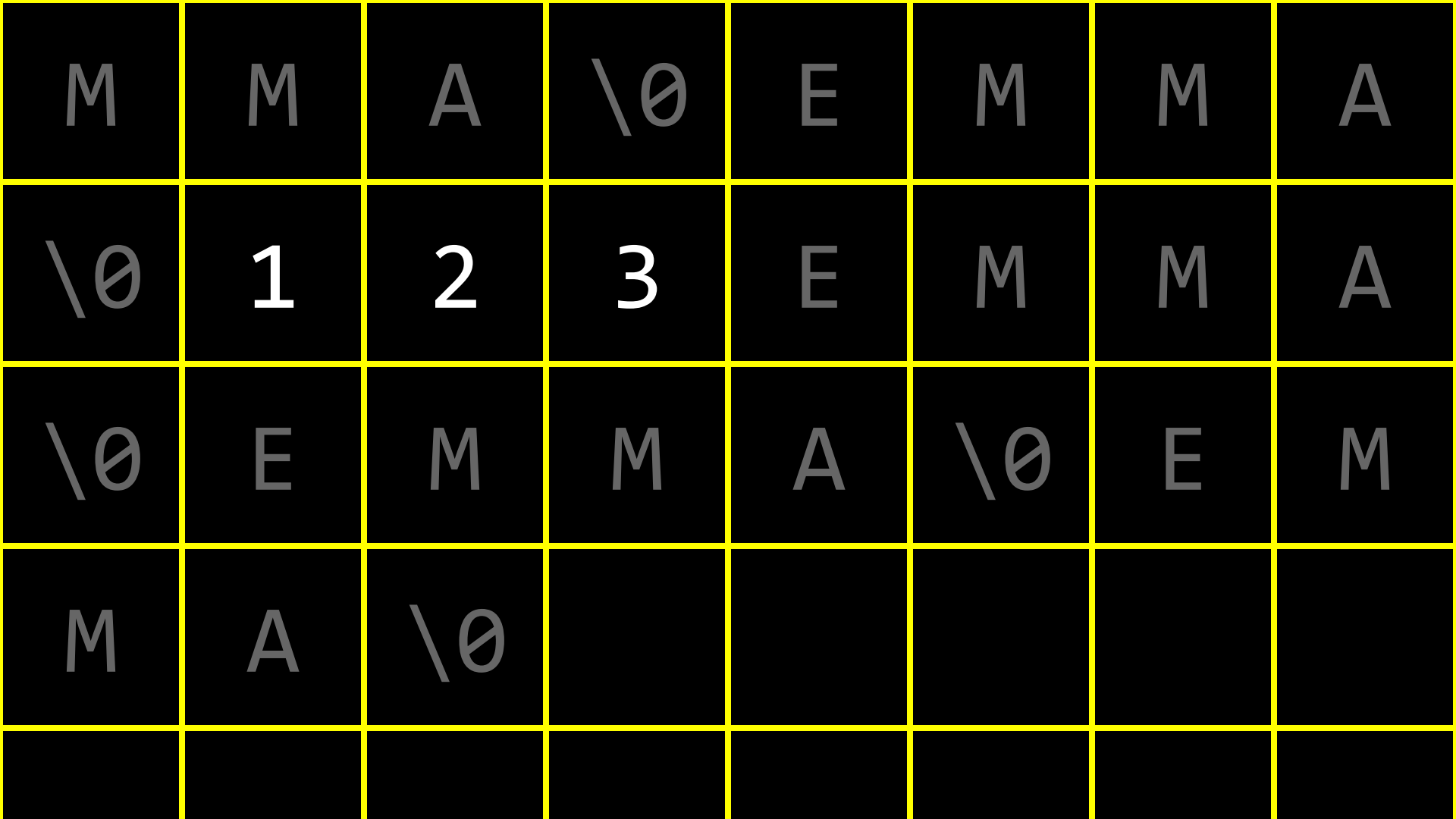
arrays

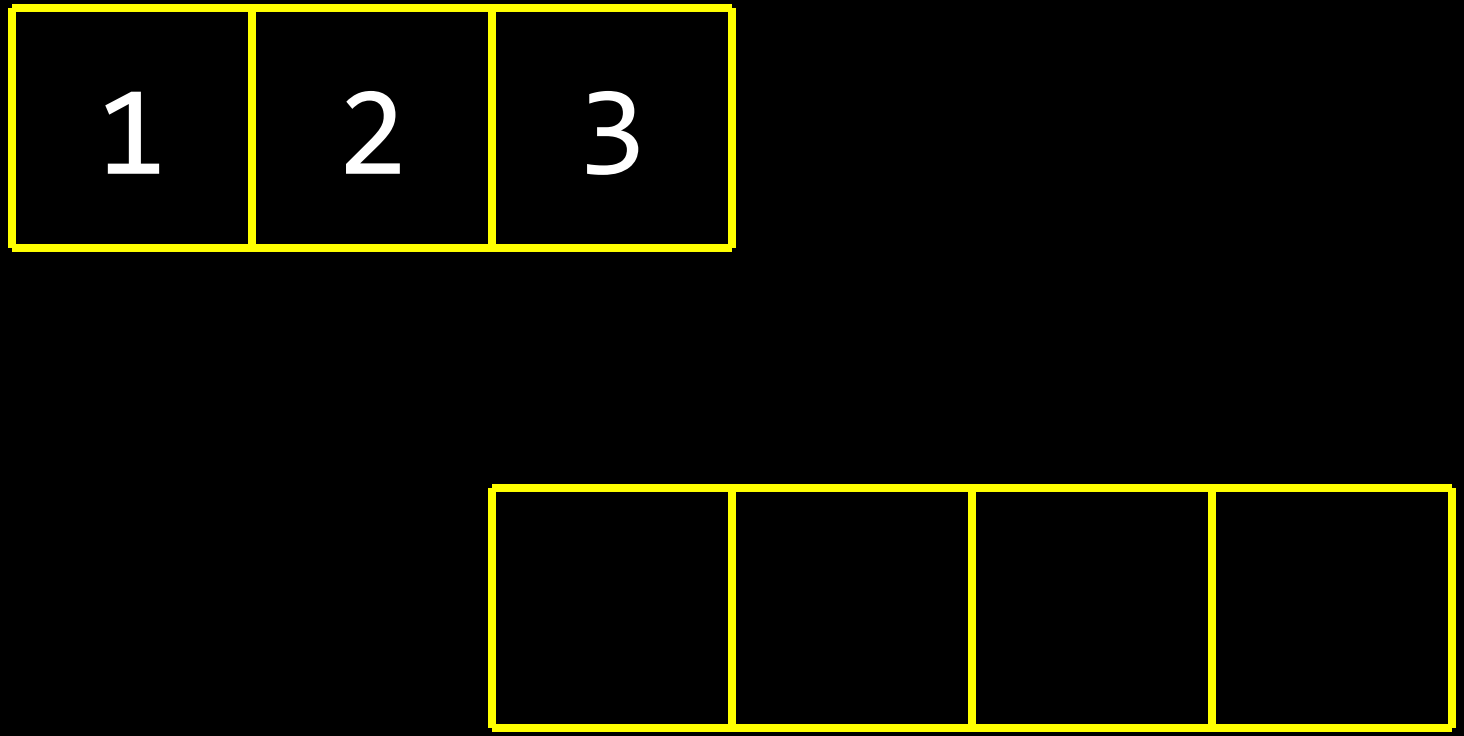


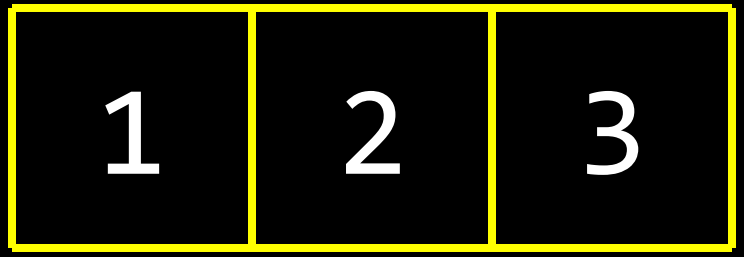

1
2
3

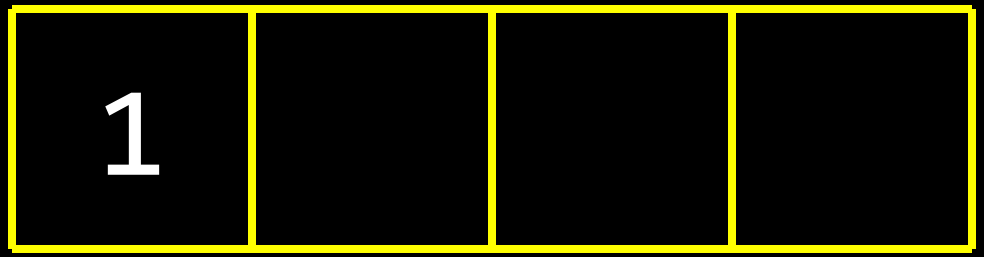

1
2
3

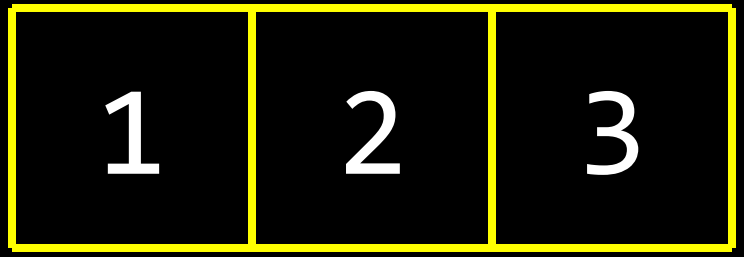
1 2 3

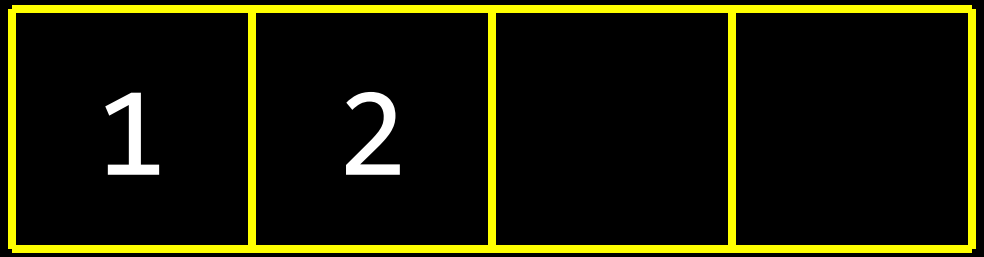

M M A \0 E M M A
\0 1 2 3 E M M A
\0 E M M A \0 E M
M A \0

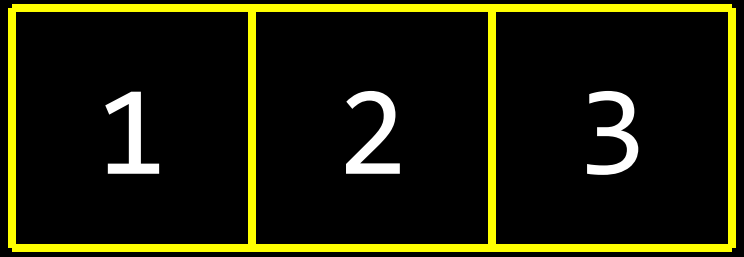

1
2
3

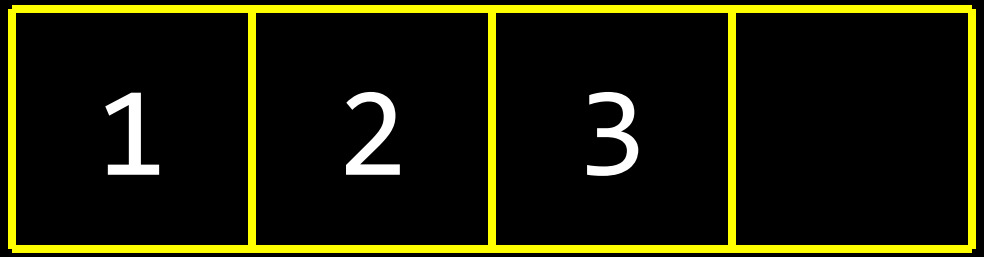

1
2
3

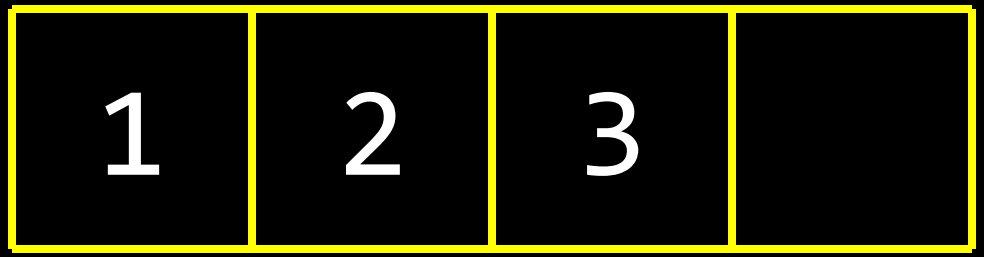

1

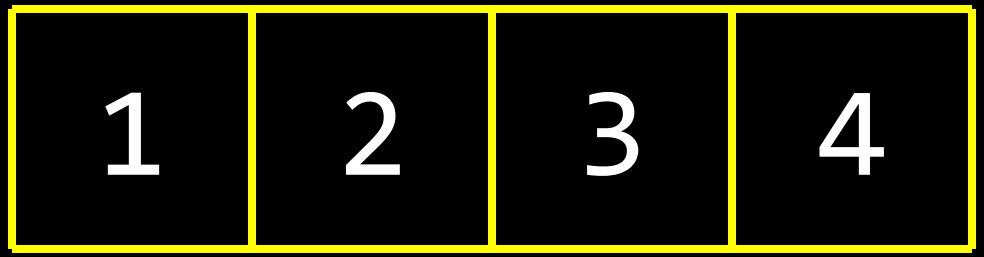

1
2
3


1
2


1
2
3


1
2
3


1
2
3


1
2
3
4

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$ search

$O(1)$

$O(n^2)$

$O(n \log n)$

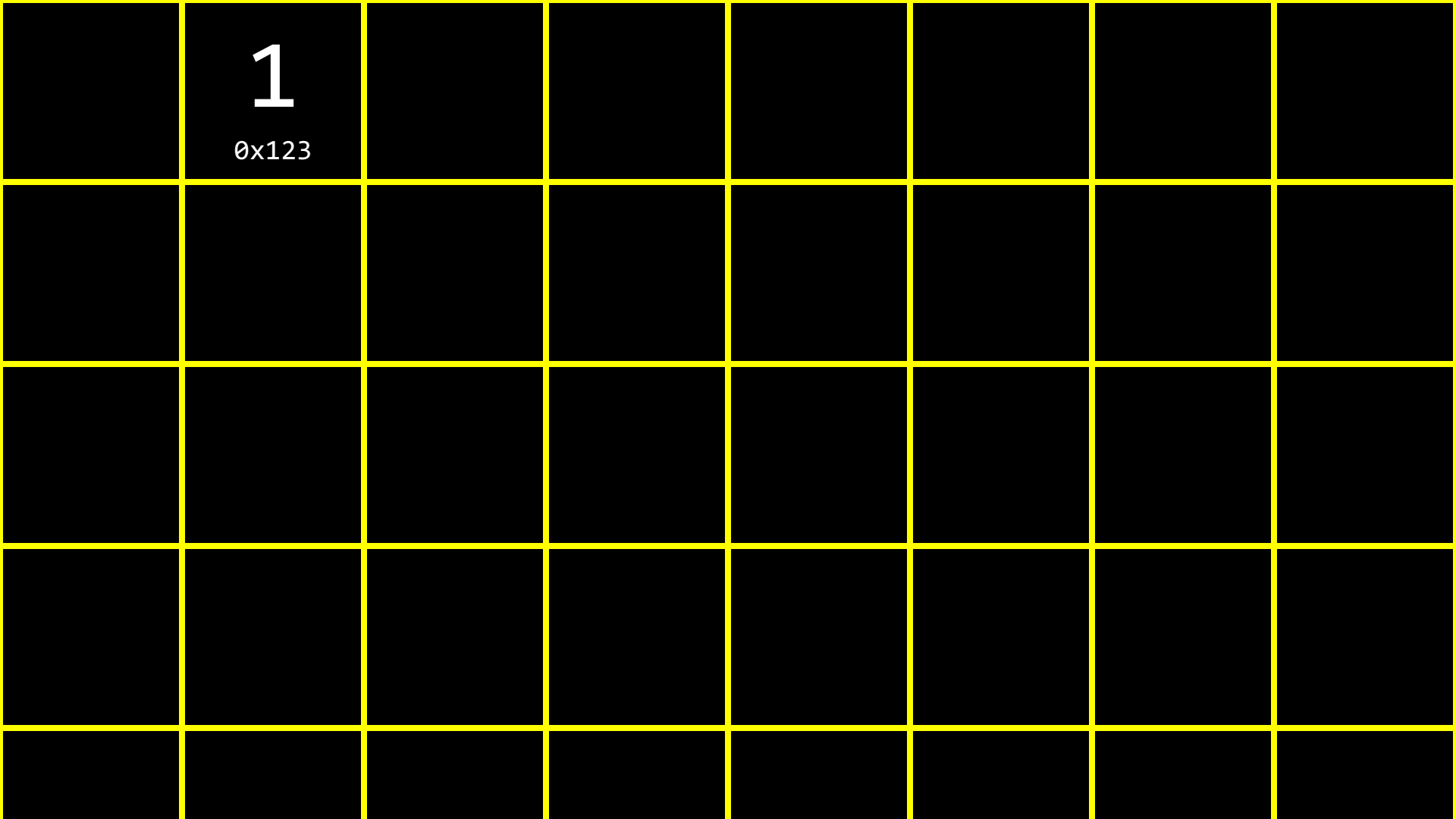
$O(n)$ insert

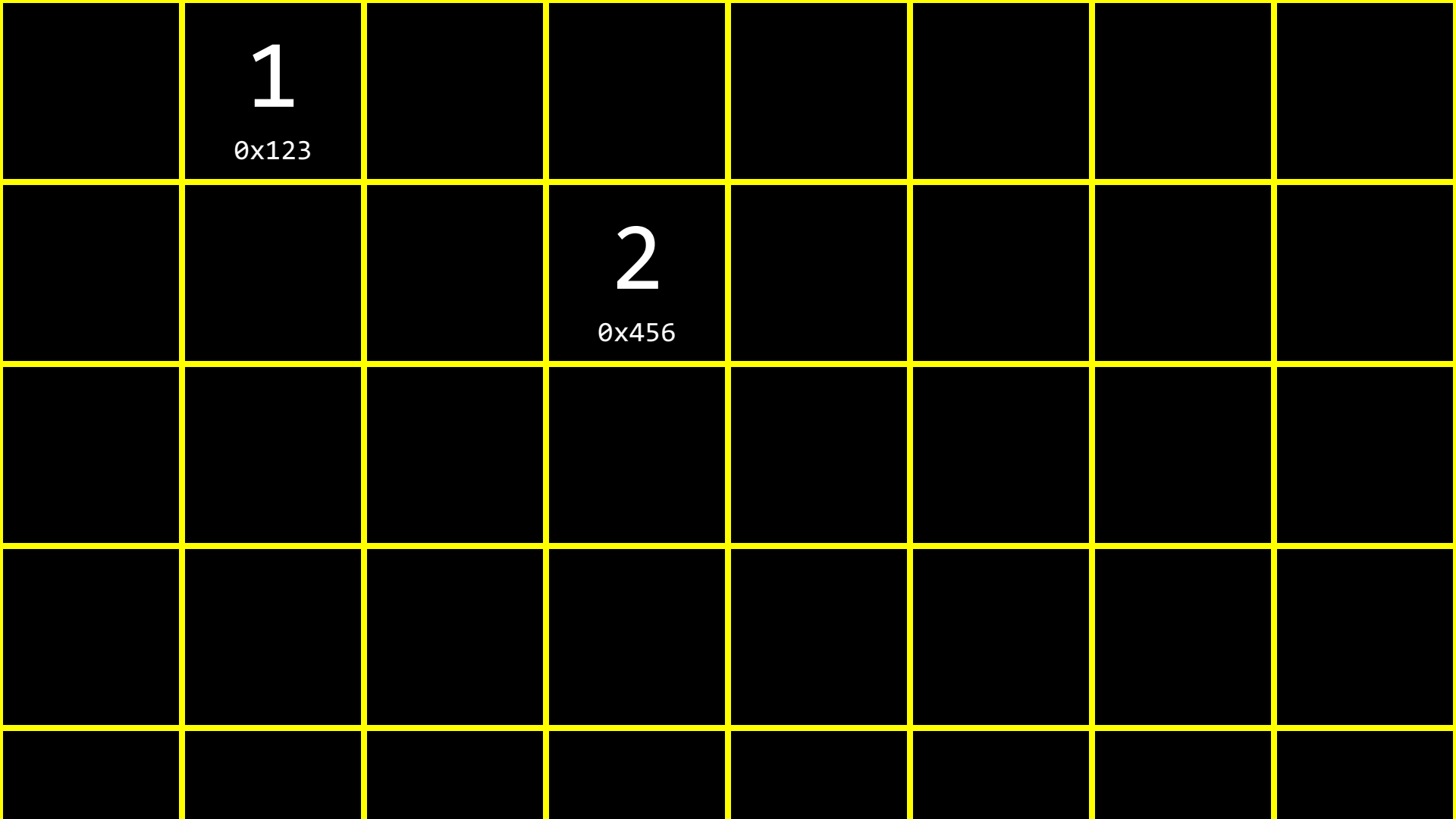
$O(\log n)$ search

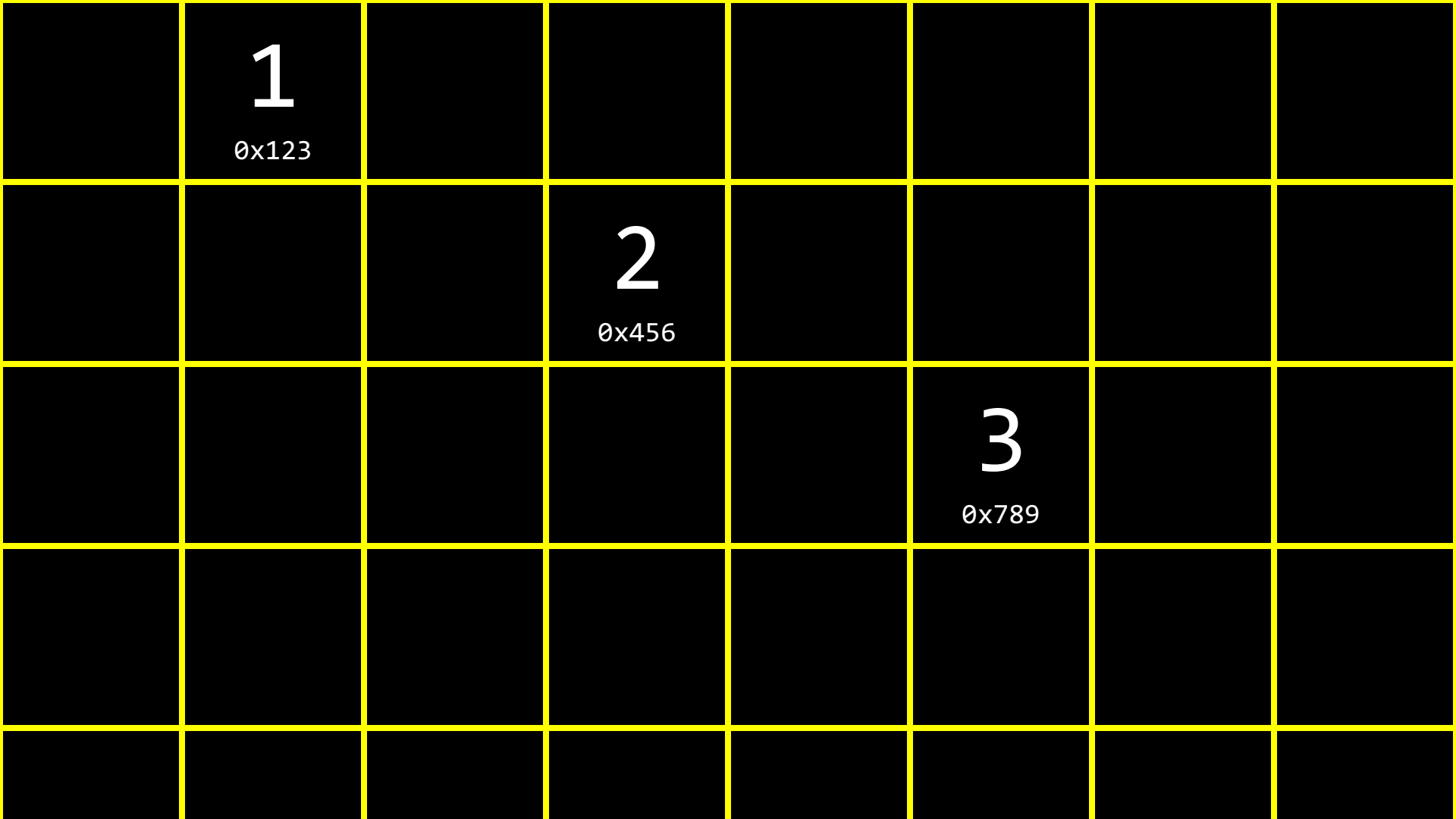
$O(1)$

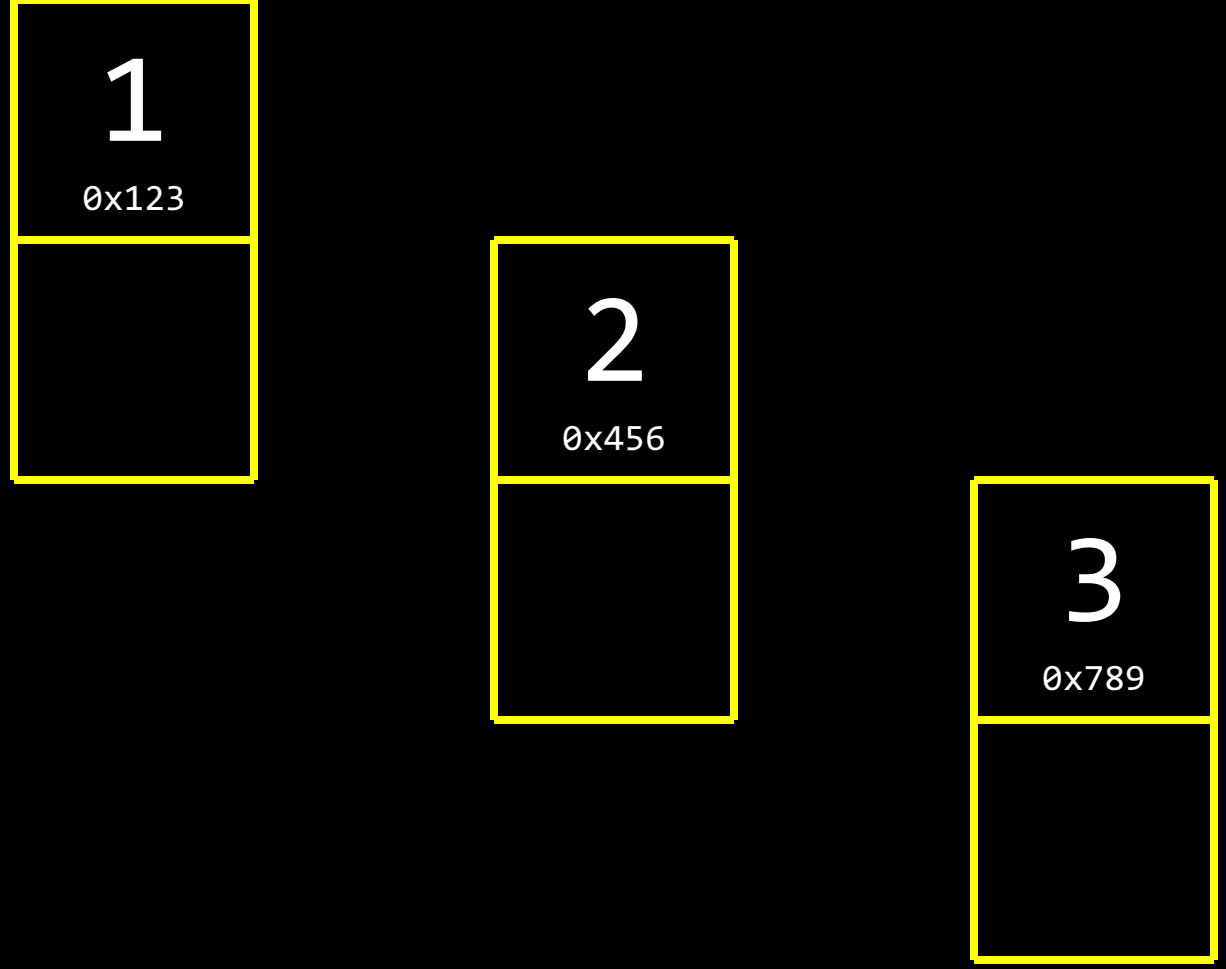
data structures

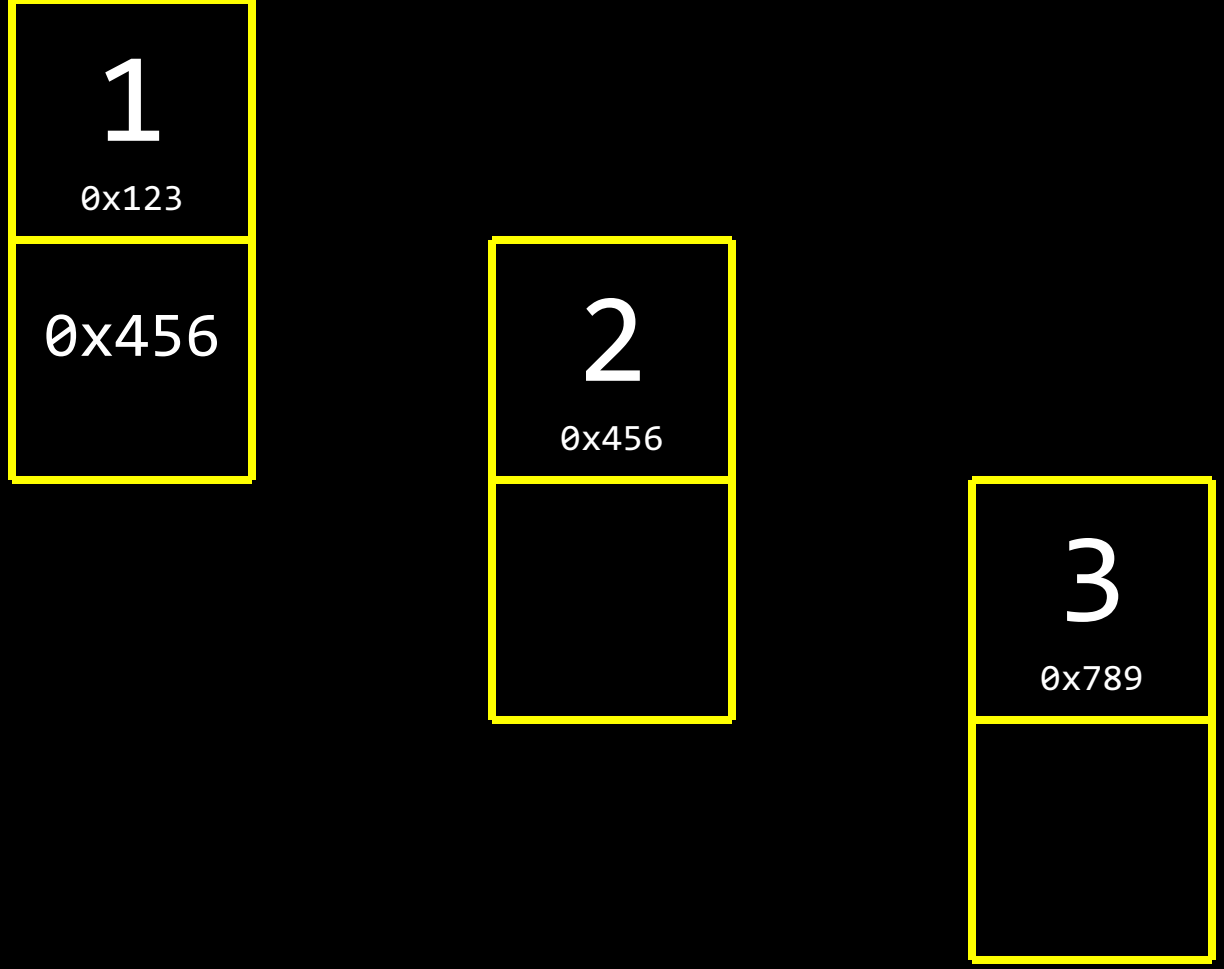
linked lists

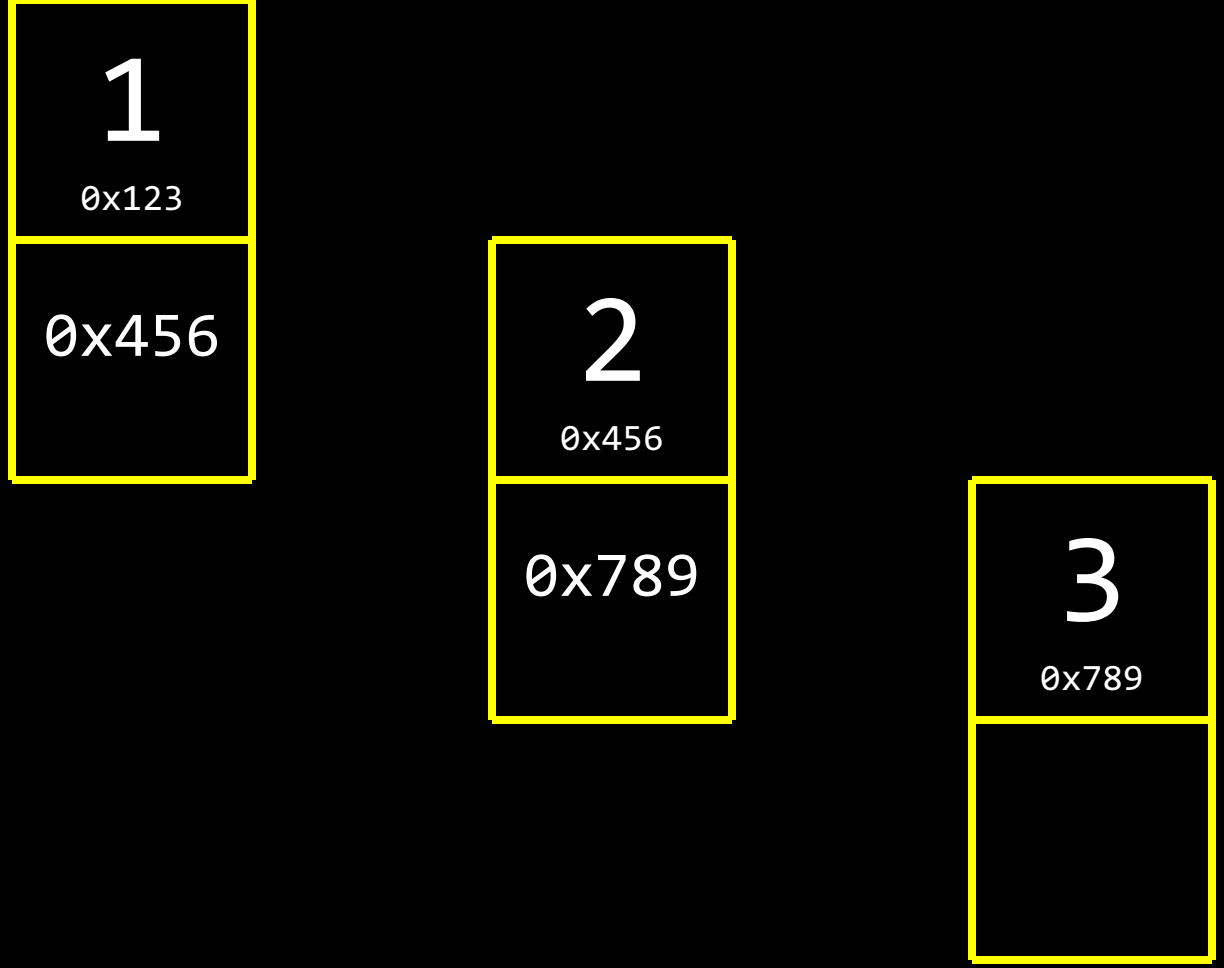

1
0x123

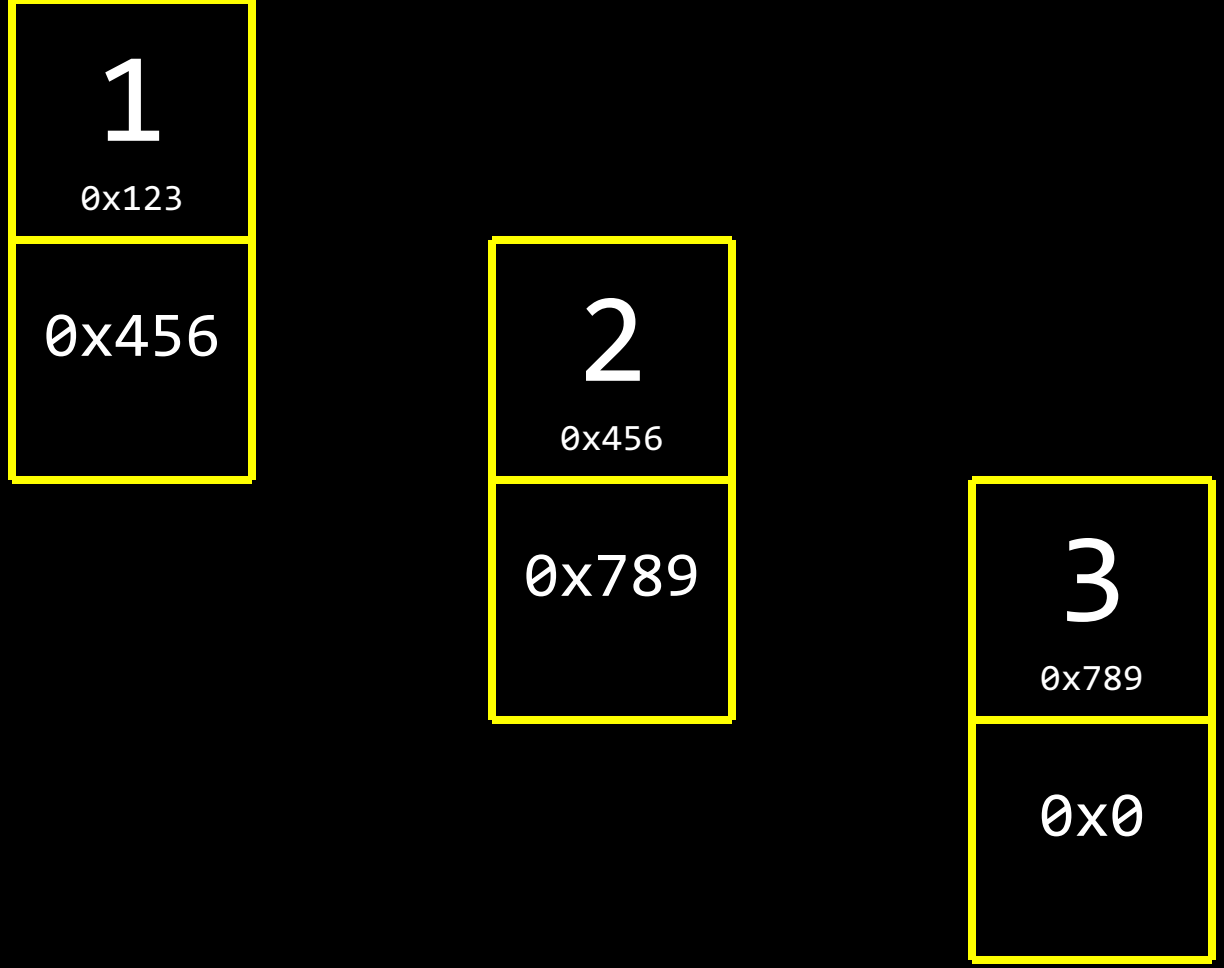

1
0x123
2
0x456

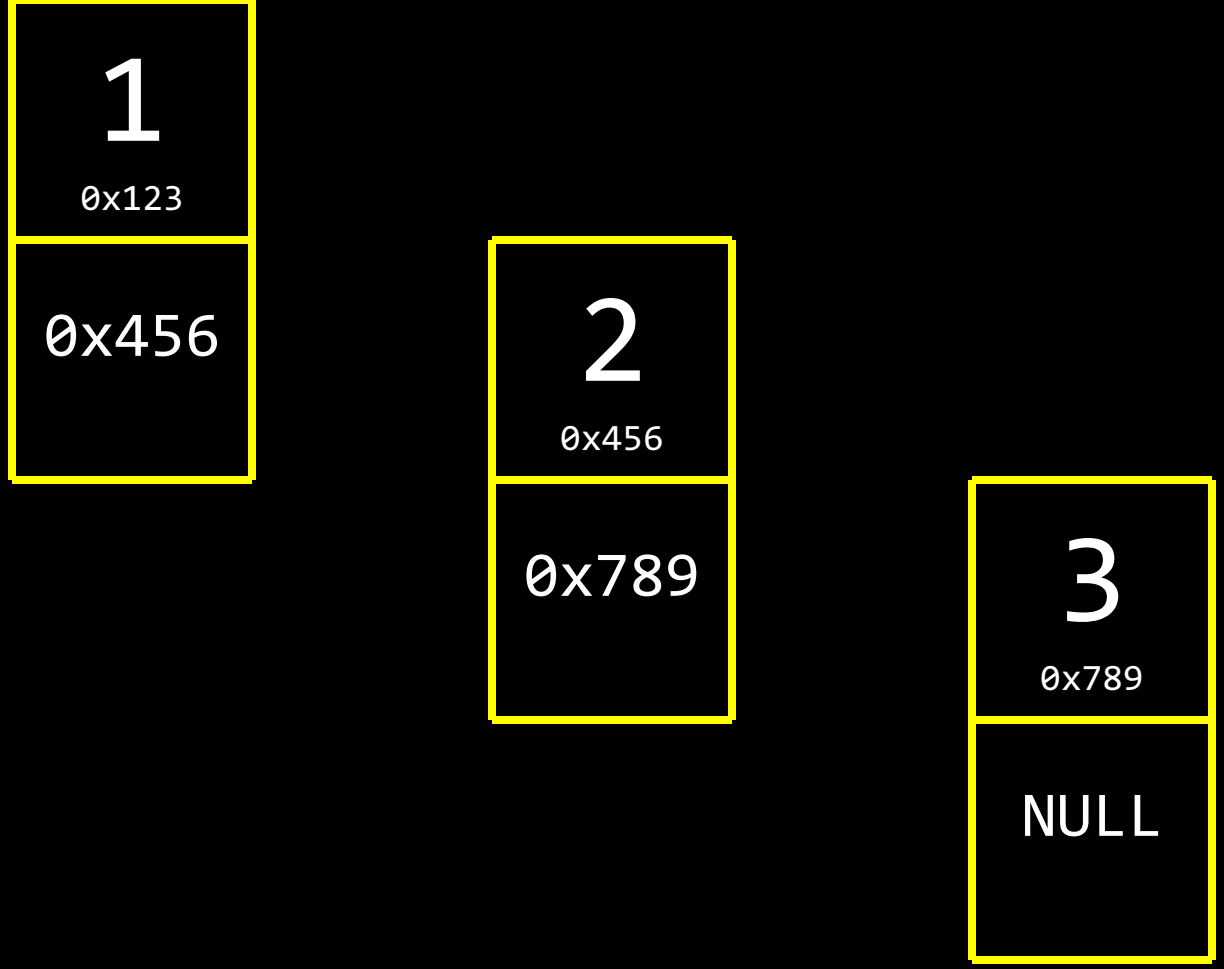

1
0x123
2
0x456
3
0x789

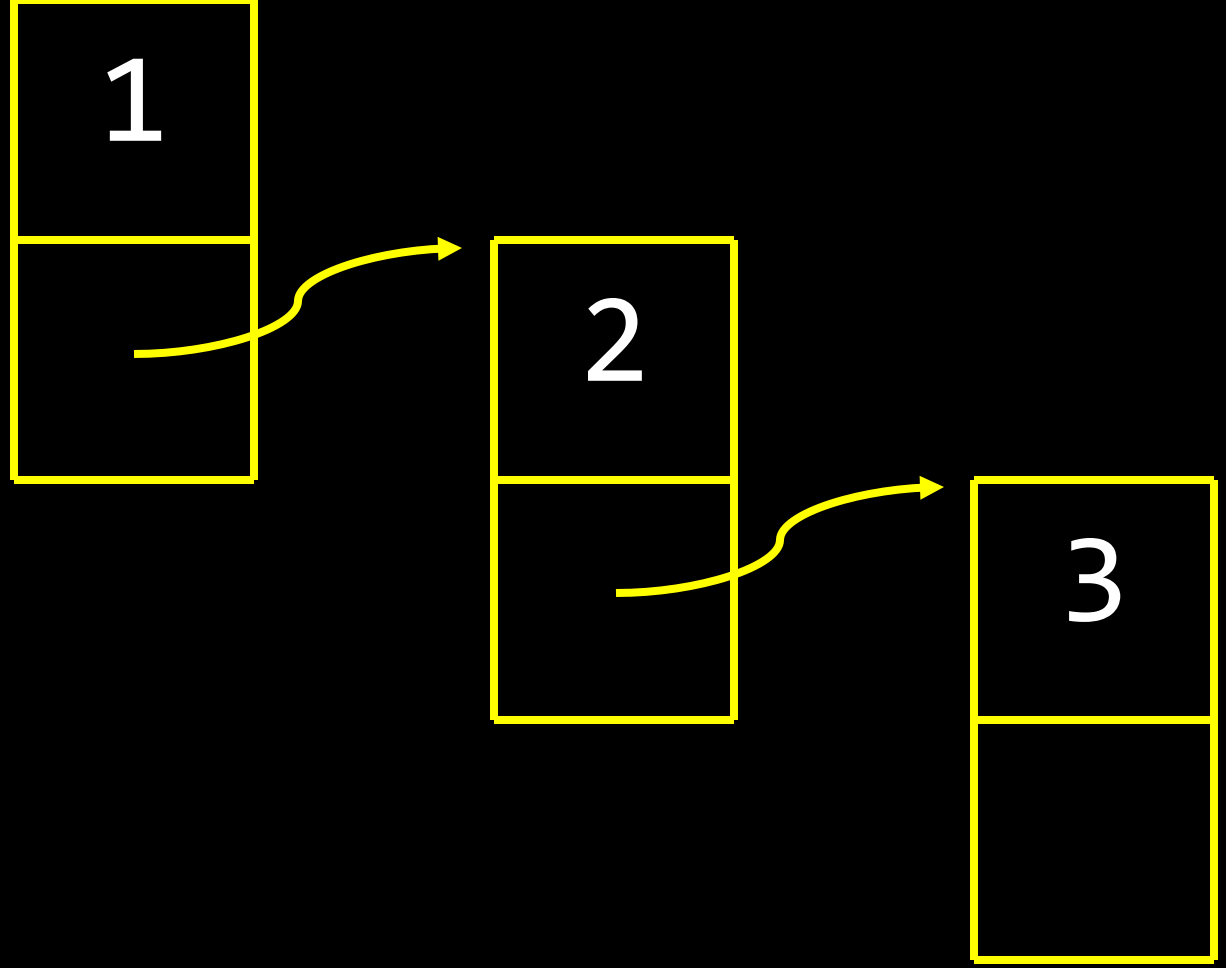

1
0x123
2
0x456
3
0x789


1
0x123
0x456
2
0x456
3
0x789


1
0x123
0x456
2
0x456
0x789
3
0x789


1
0x123
0x456
2
0x456
0x789
3
0x789
0x0


1
0x123
0x456
2
0x456
0x789
3
0x789
NULL


1
2
3

```
Class Node
{


}
```

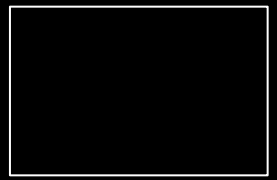
```
Class Node
{
    int number;

}
```

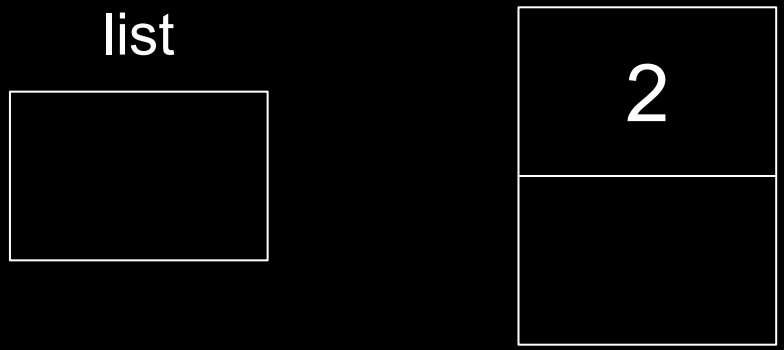
```
Class Node
{
    int number;
    Node next;
}
```

The first and the last node of a linked list are known as the head and tail of the list respectively.

```
Node list = NULL;
```

list

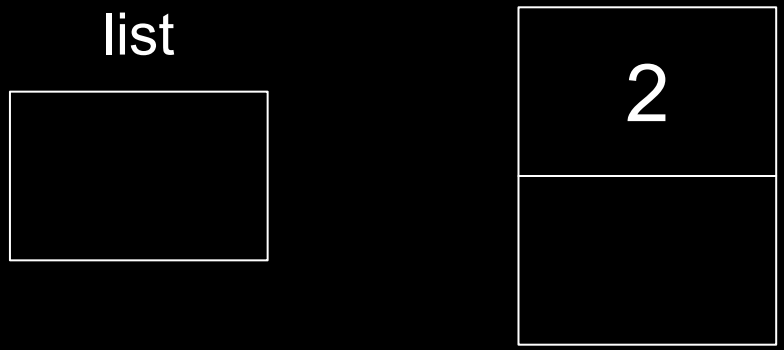


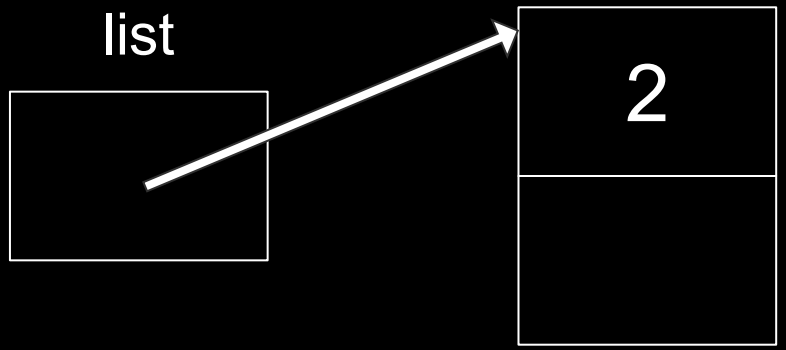

list
2

```
number = 2;
```

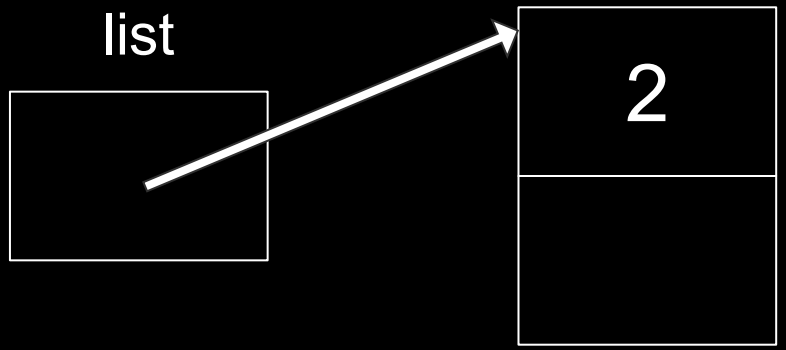
```
number = 2;
next = NULL;
```

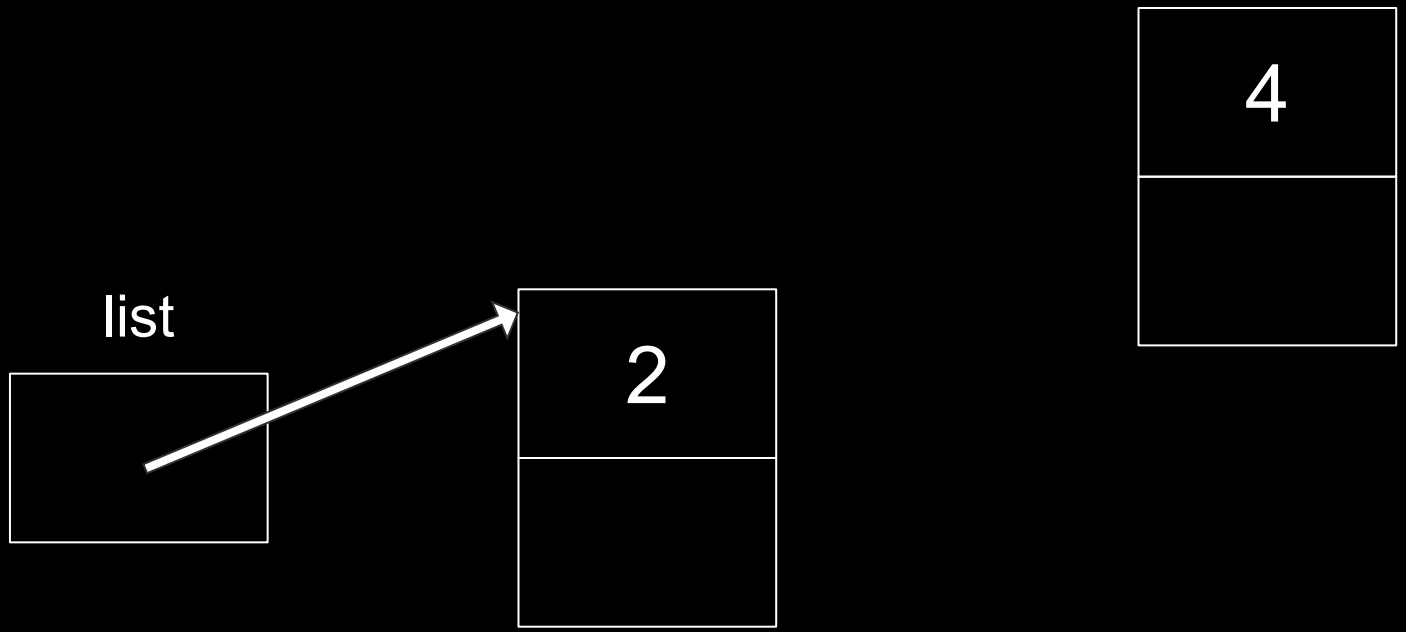
```
if (n != NULL)
{
   number = 2;
   next = NULL;
}
```


list
2

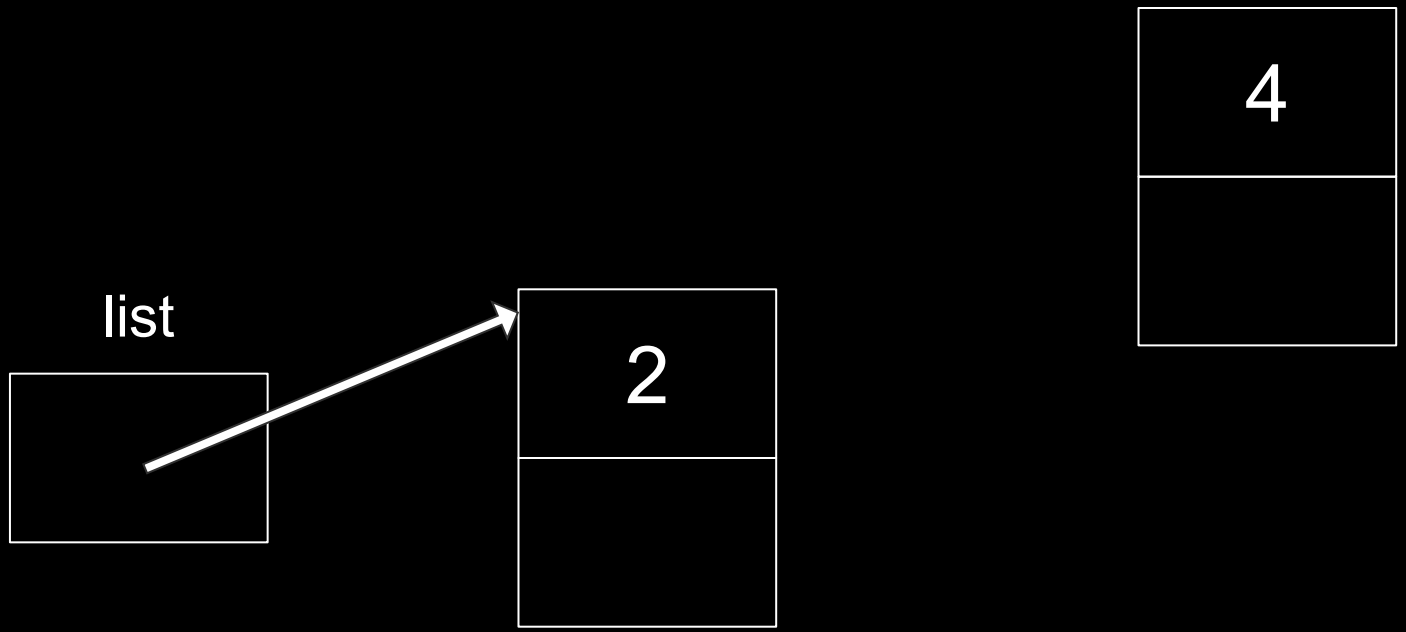

list
2

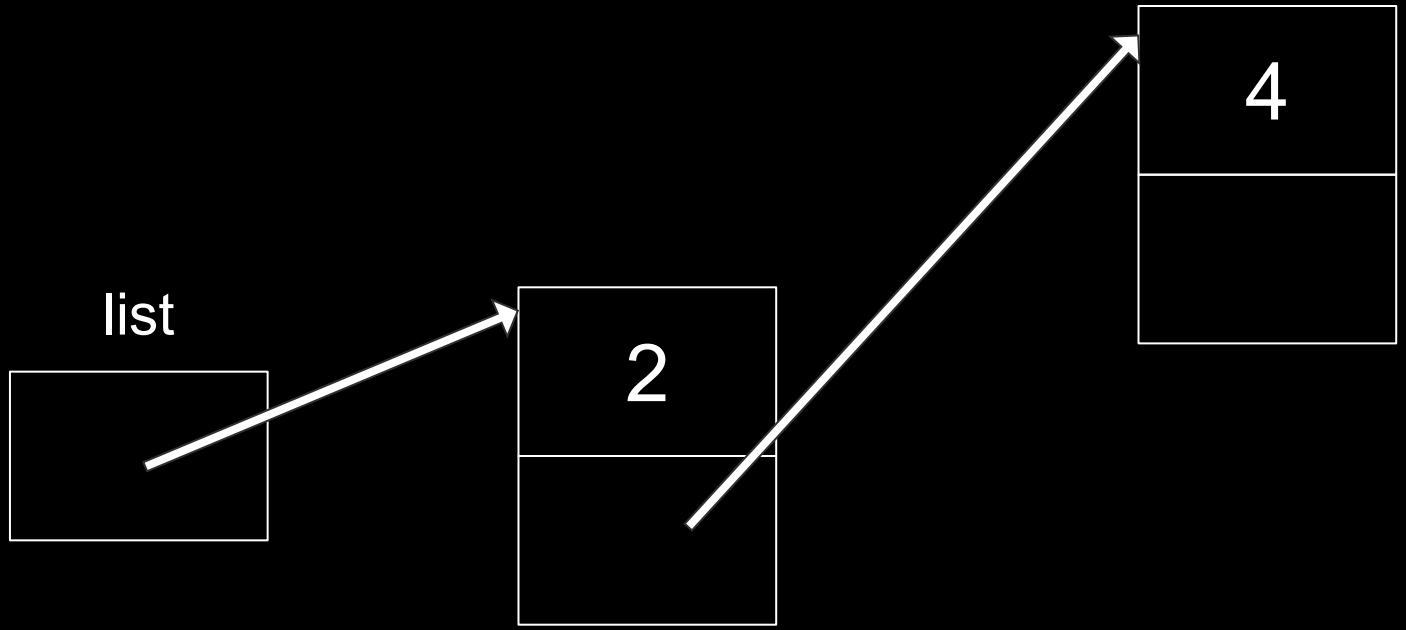
```
list = n;
```


list
2


4
list
2

```
if (n != NULL)
{
    number = 4;
    next = NULL;
}
```


4
list
2

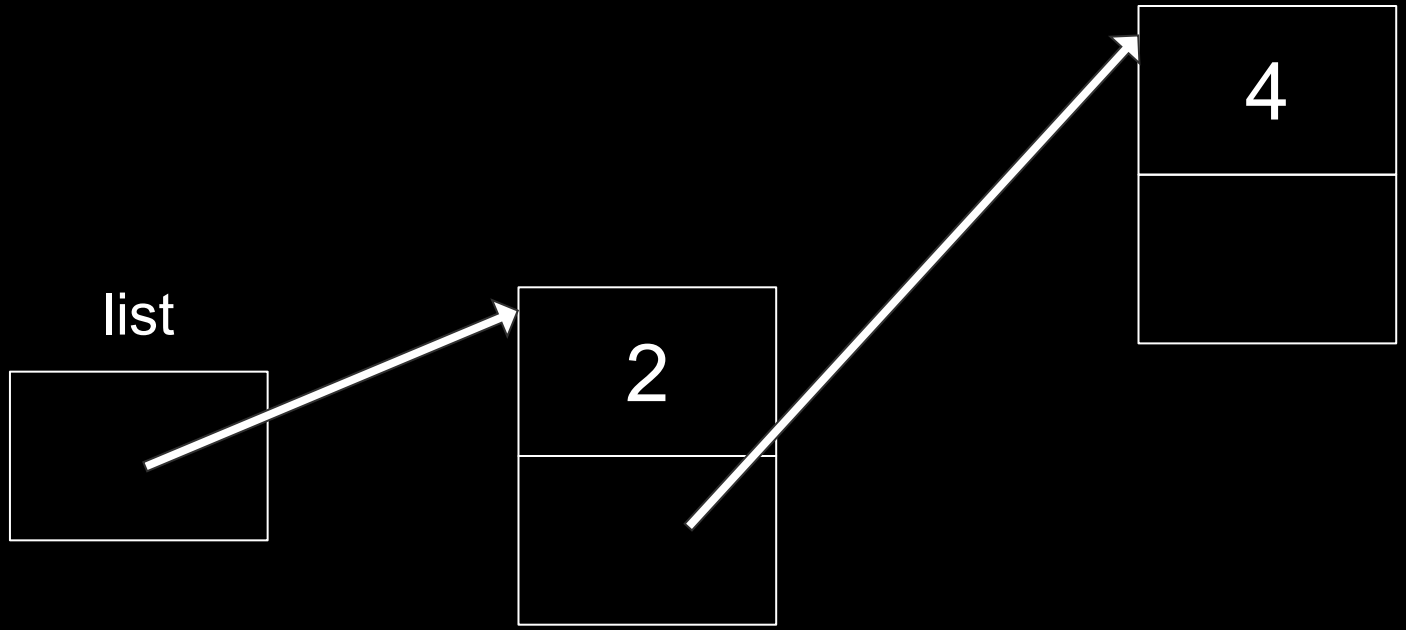

list
2
4

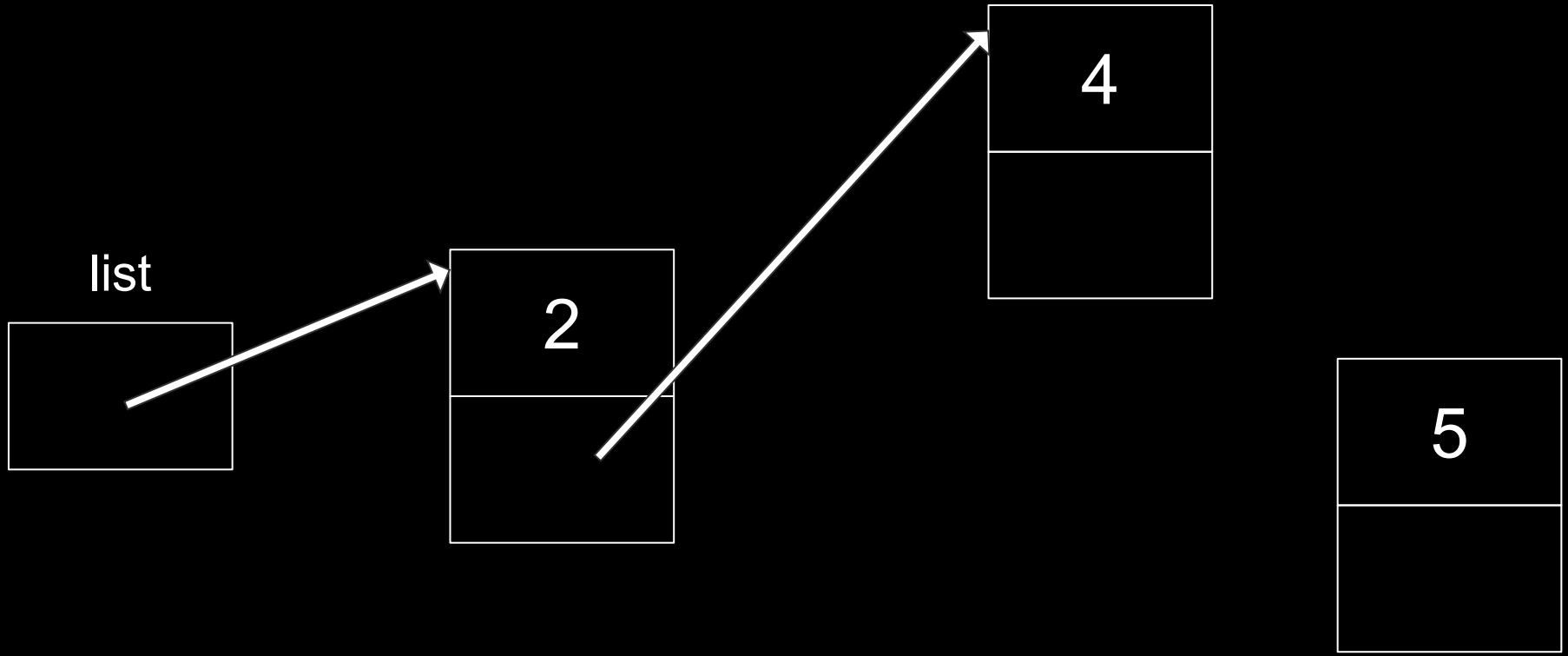
```
node tmp = list;
```

```
node tmp = list;
while (tmp.next != NULL)
{

}
```

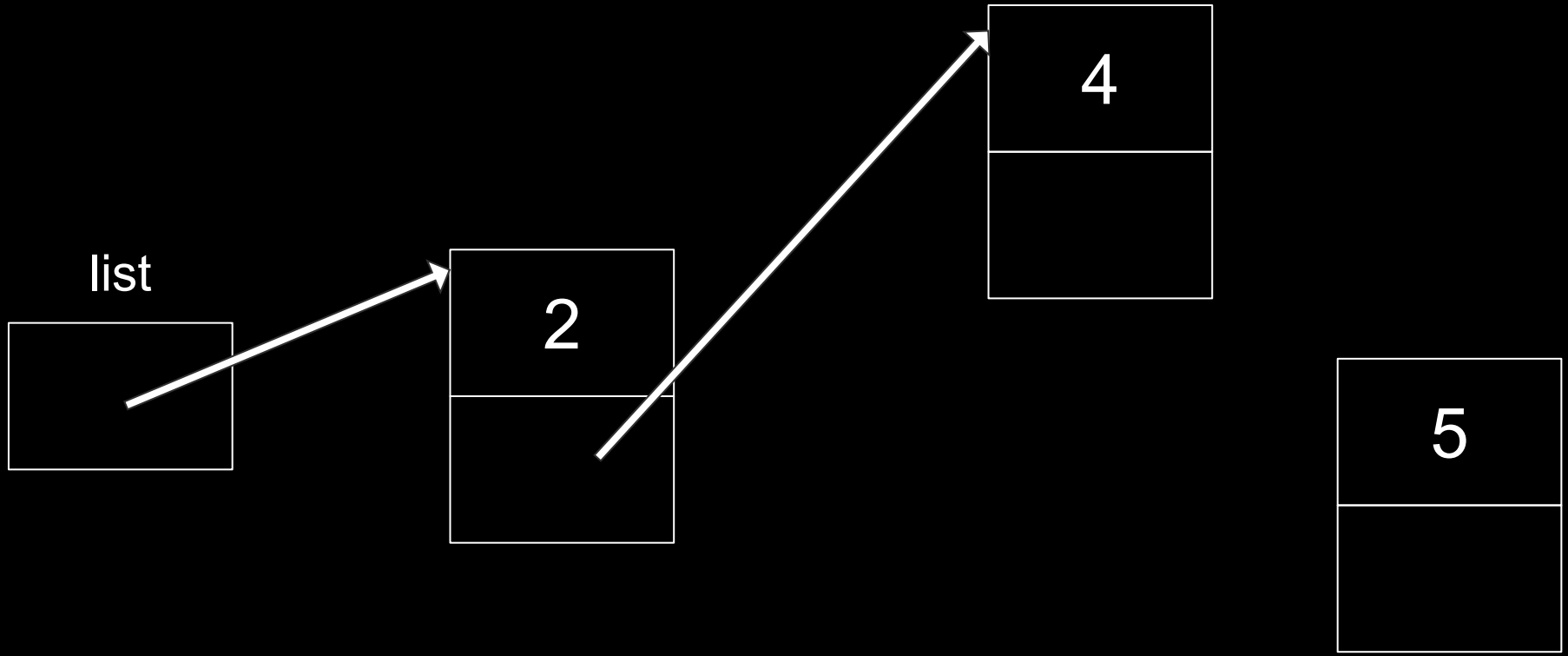
```
node tmp = list;
while (tmp.next != NULL)
{
    tmp = tmp.next;
}
```

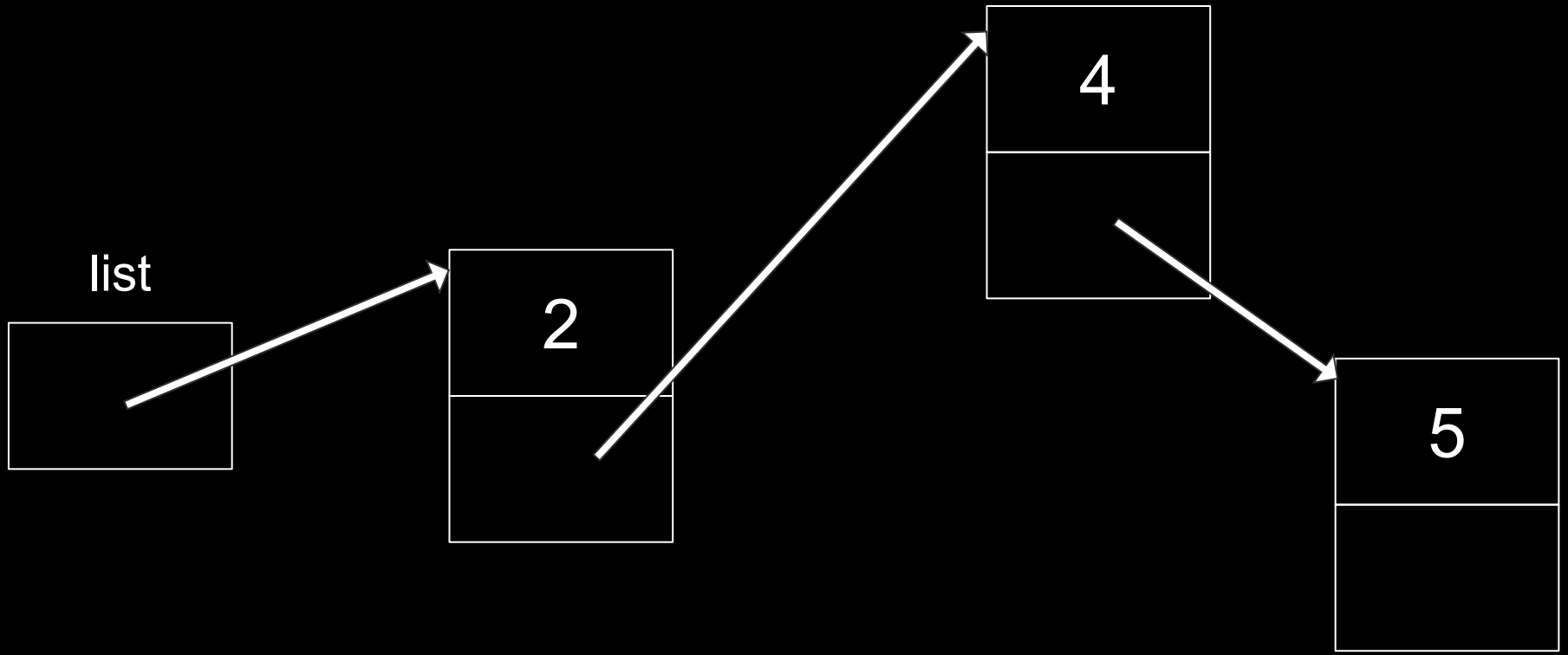
```
node tmp = list;
while (tmp.next != NULL)
{
    tmp = tmp.next;
}
tmp.next = n;
```


list
2
4

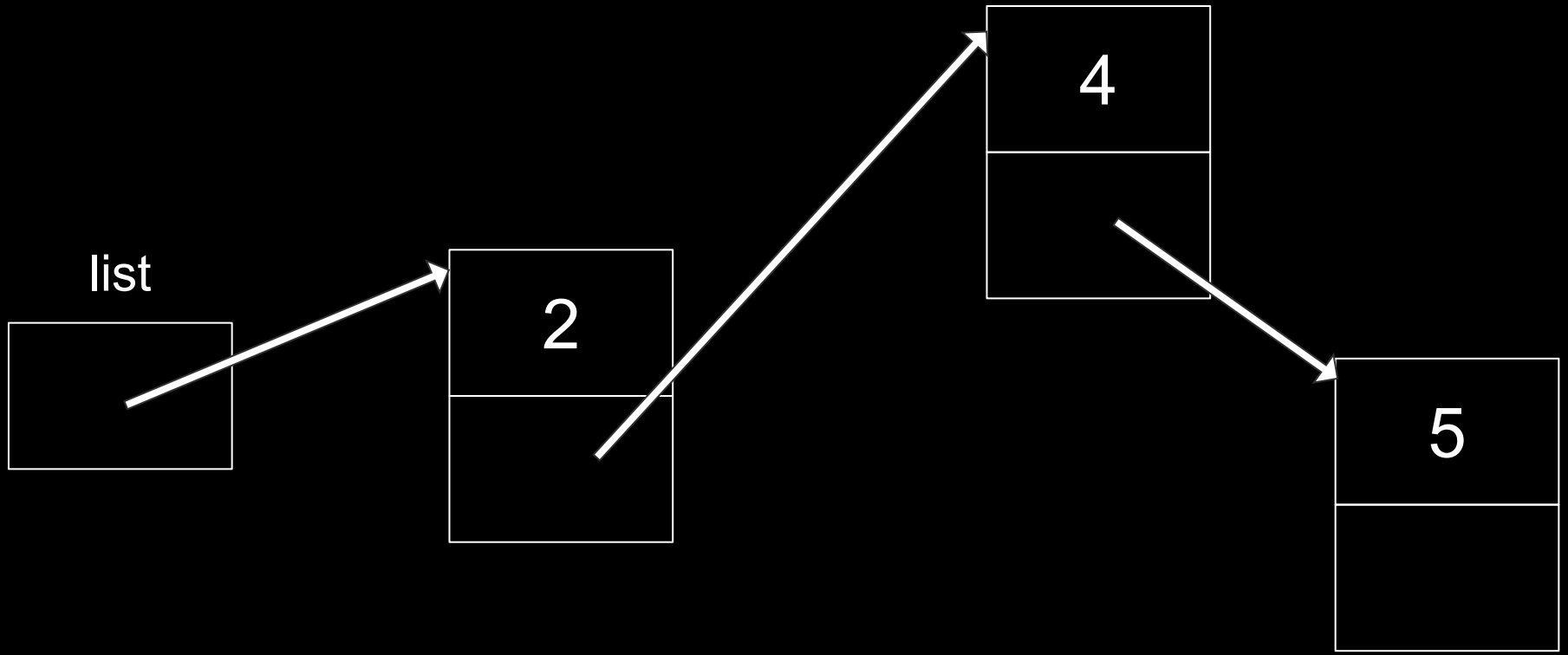

list
2
4
5

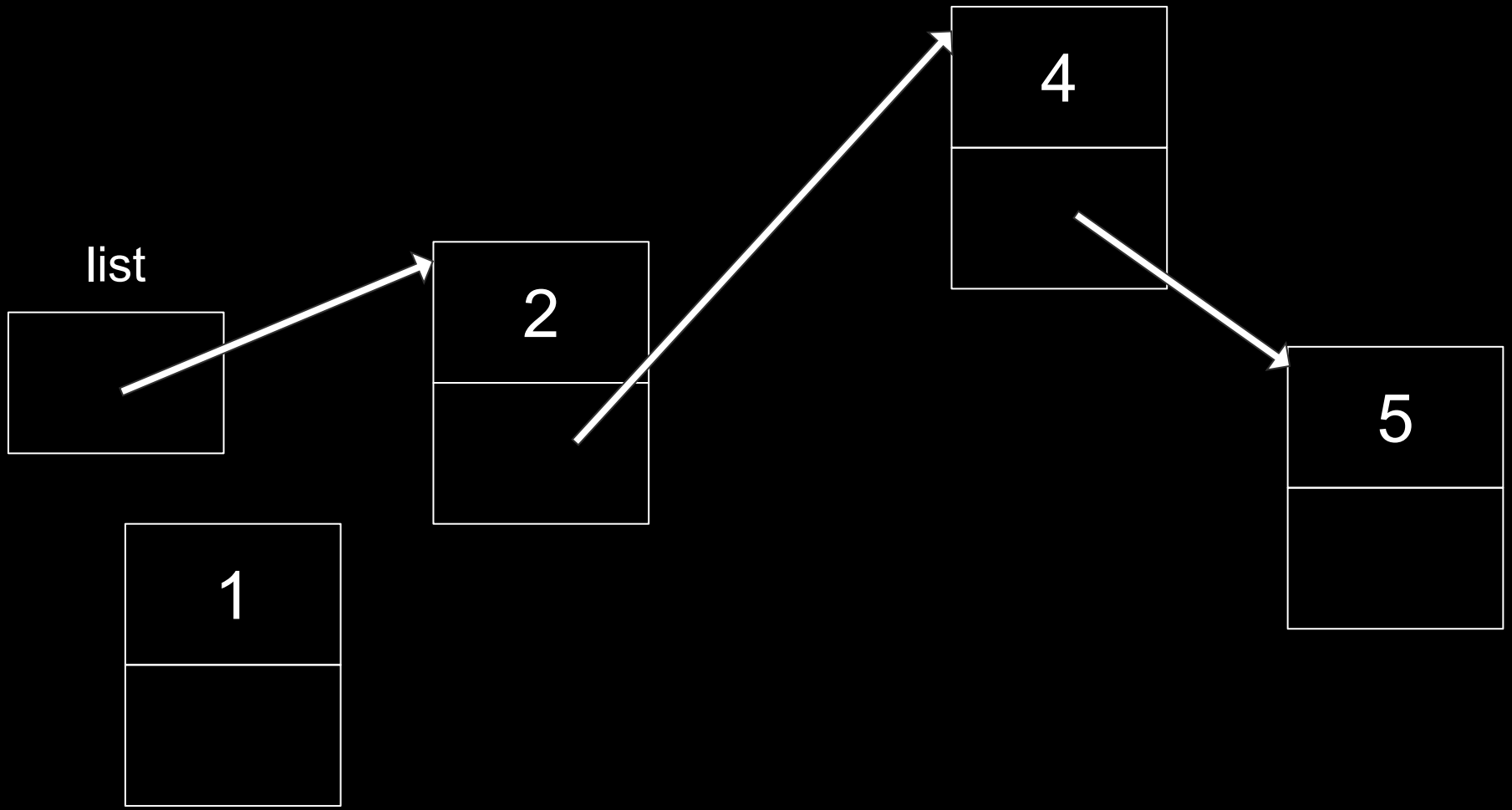
```
if (n != NULL)
{
    n.number = 5;
    n.next = NULL;
}
```


list
2
4
5

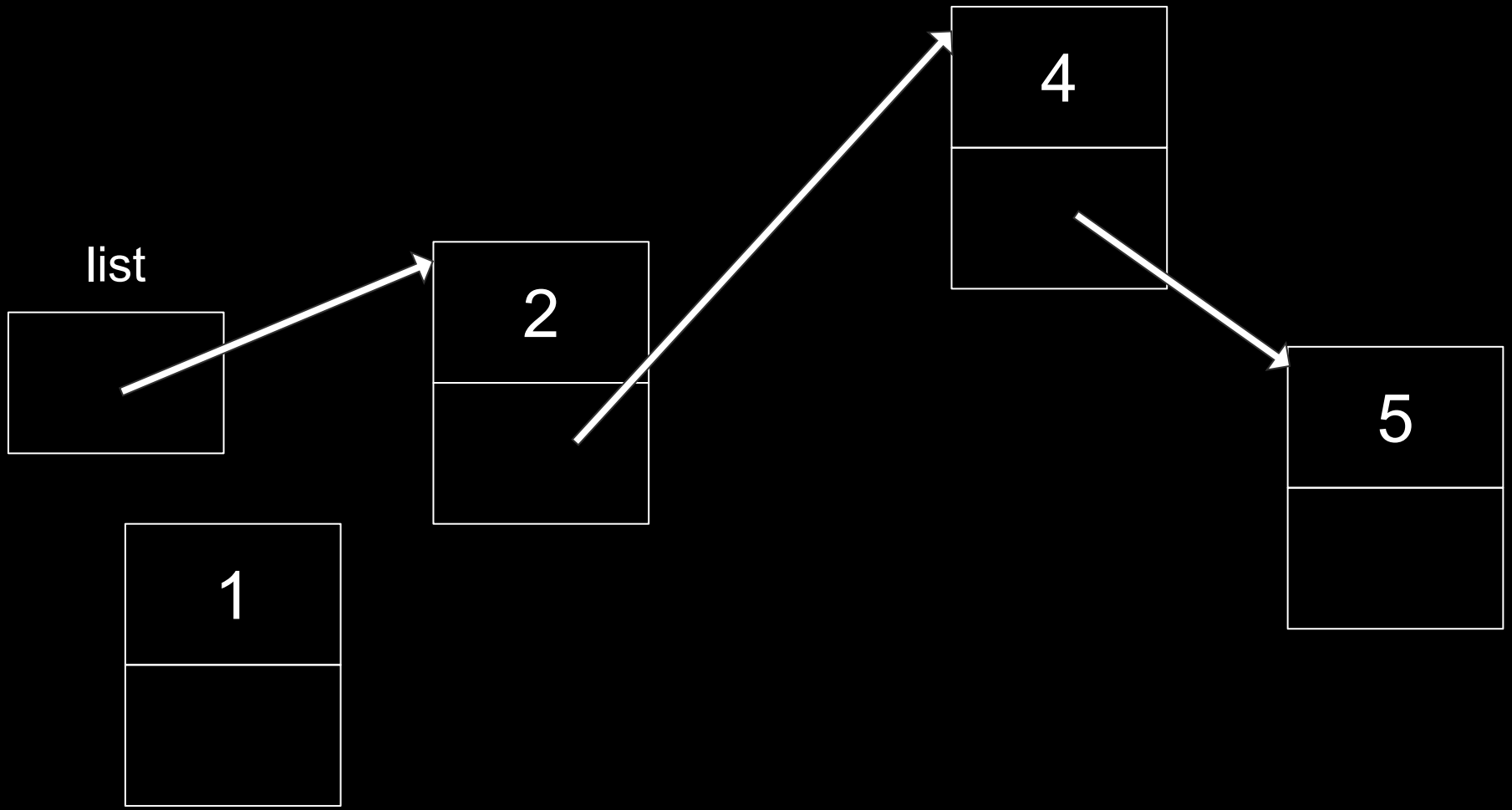

list
2
4
5

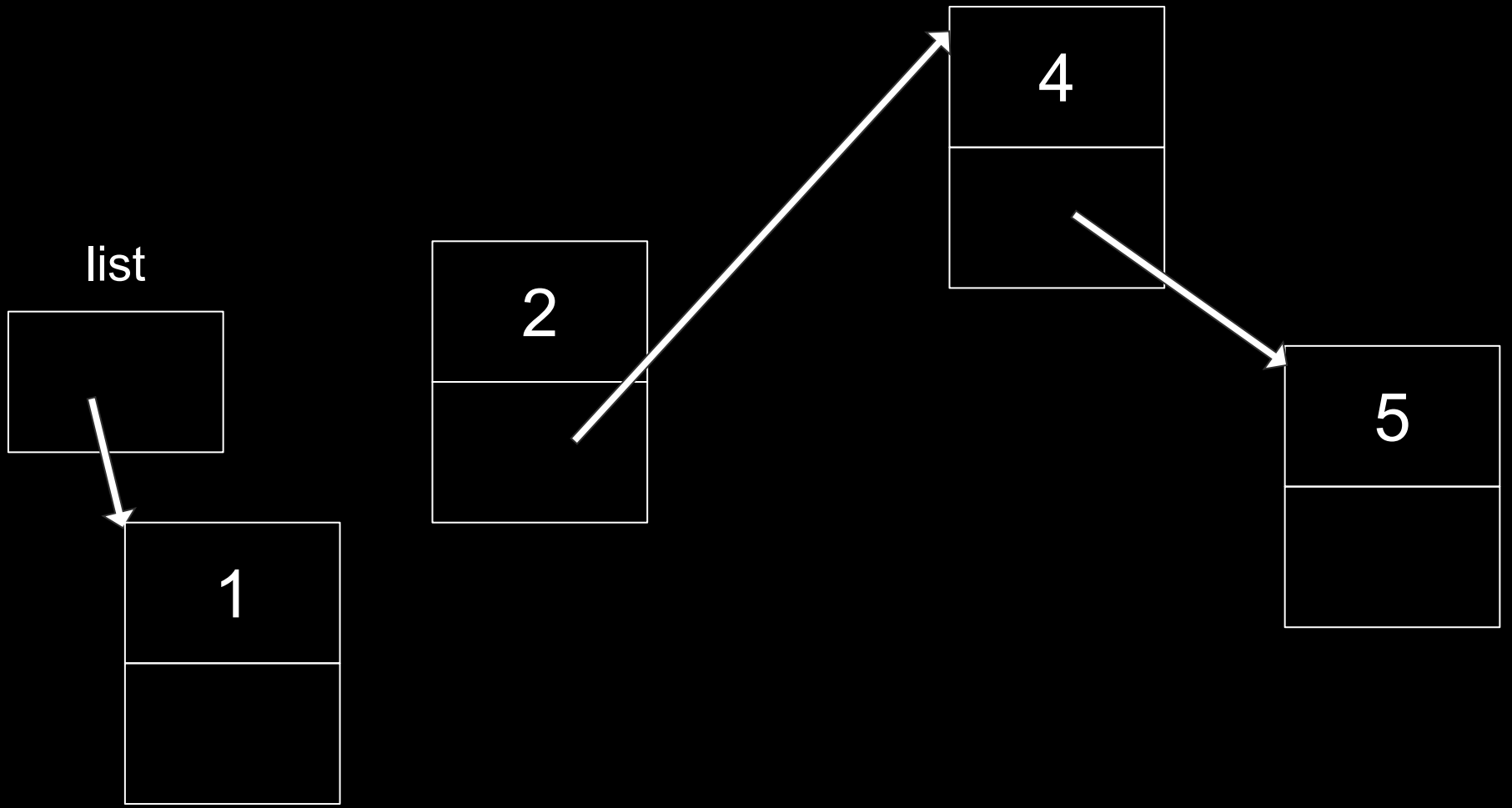
```
node tmp = list;
while (tmp.next != NULL)
{
    tmp = tmp.next;
}
tmp.next = n;
```

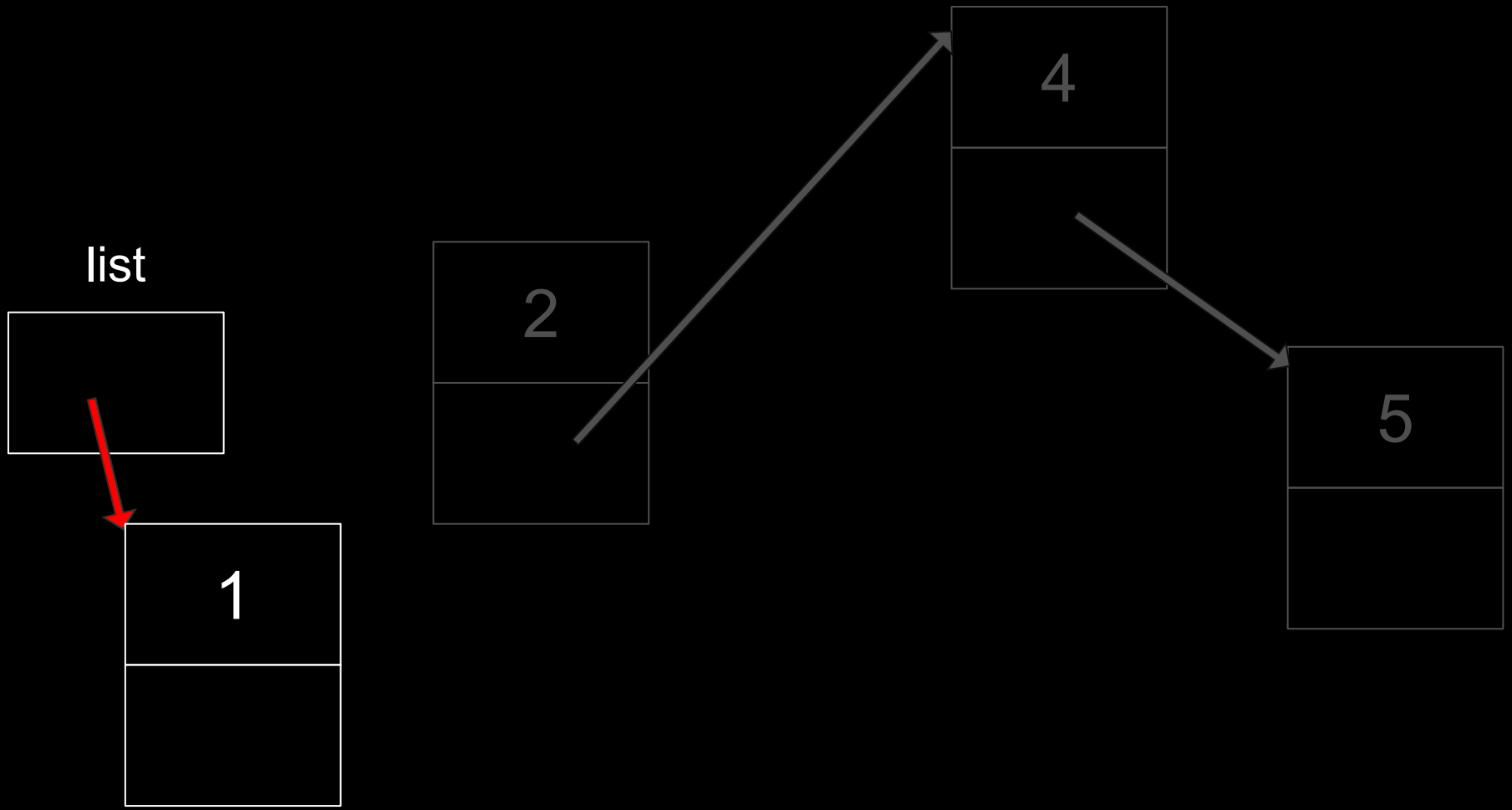

list
2
4
5

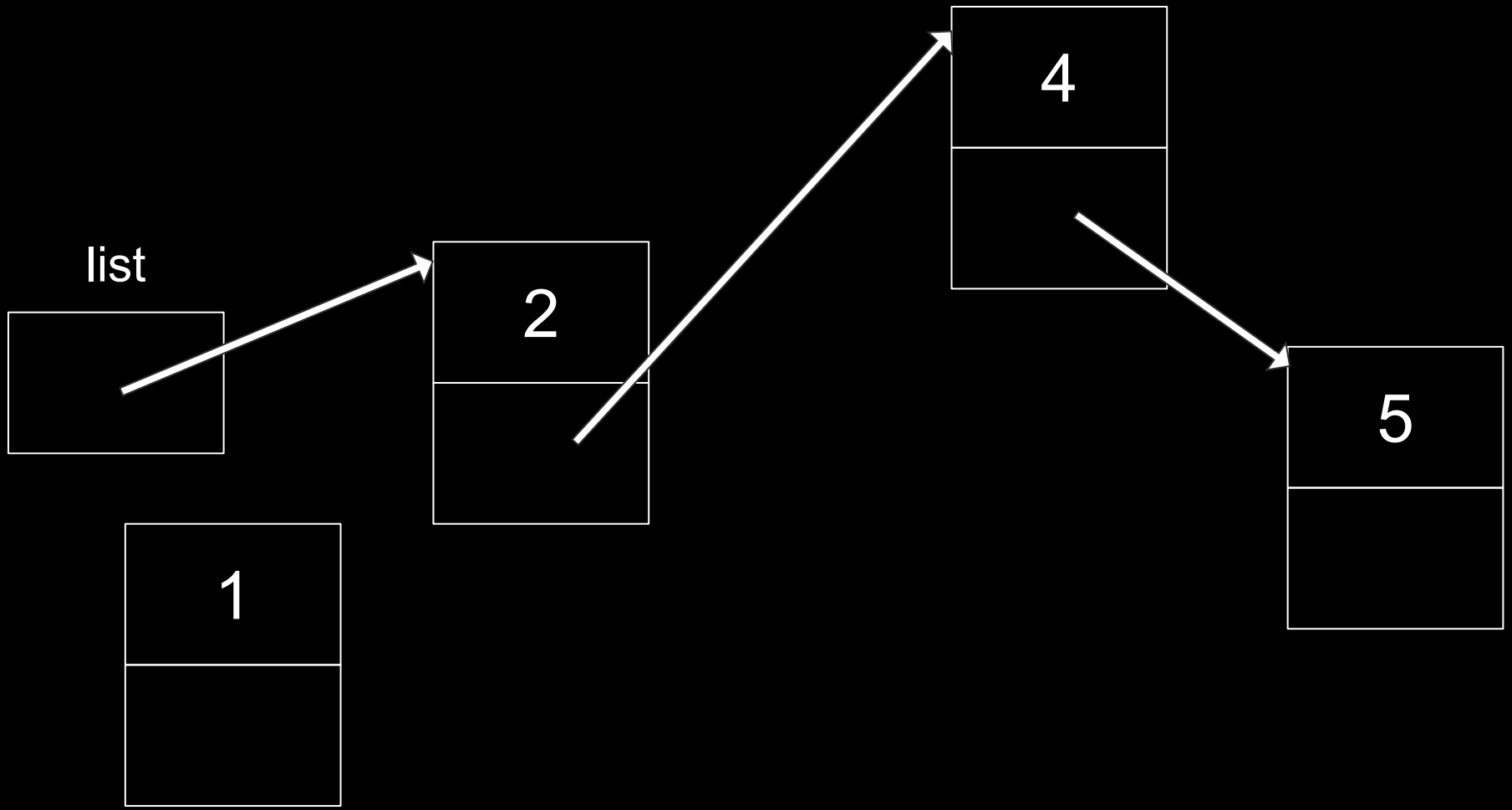

list
1
2
4
5

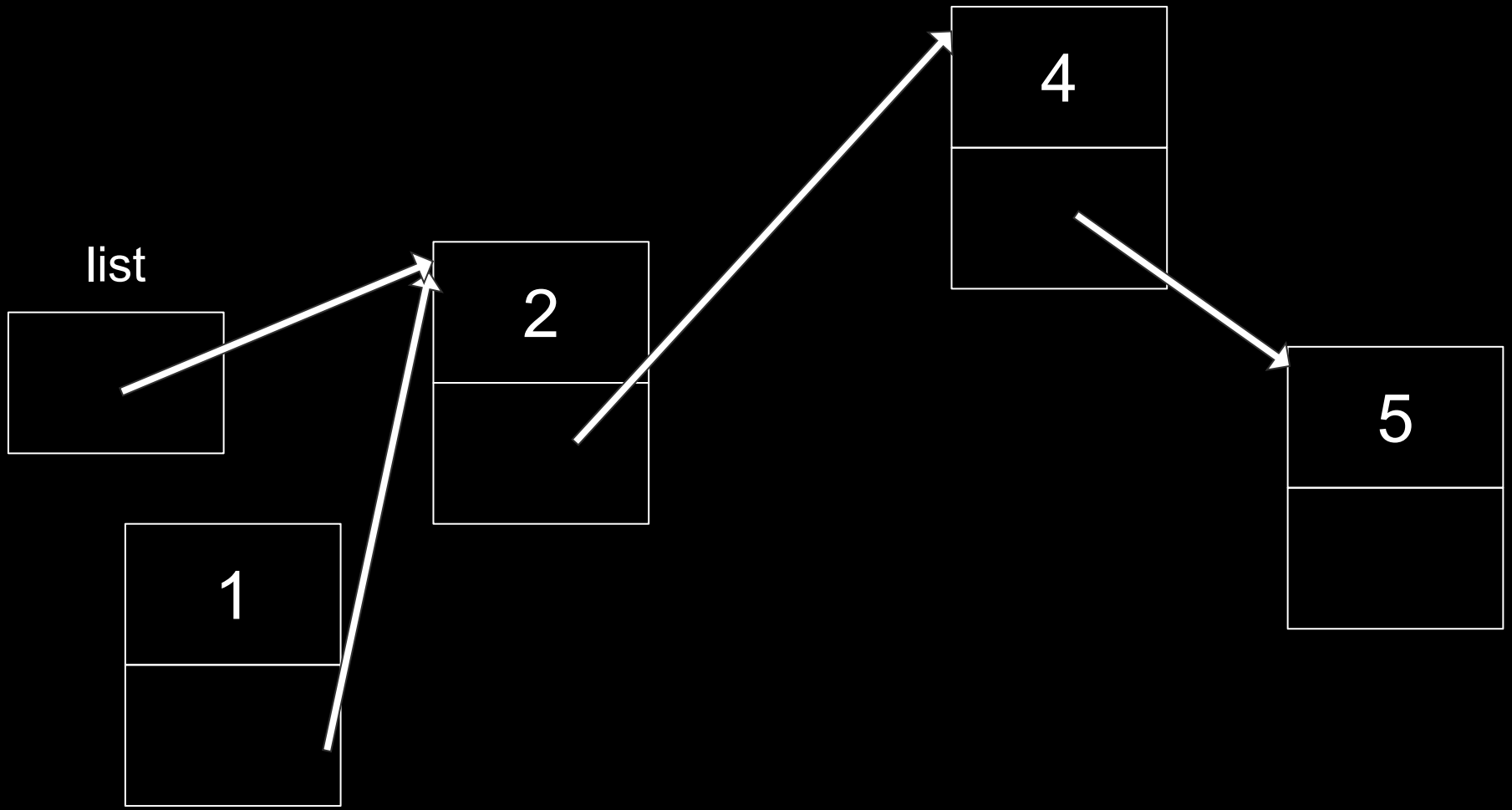
```
if (n != NULL)
{
    n.number = 1;
    n.next = NULL;
}
```

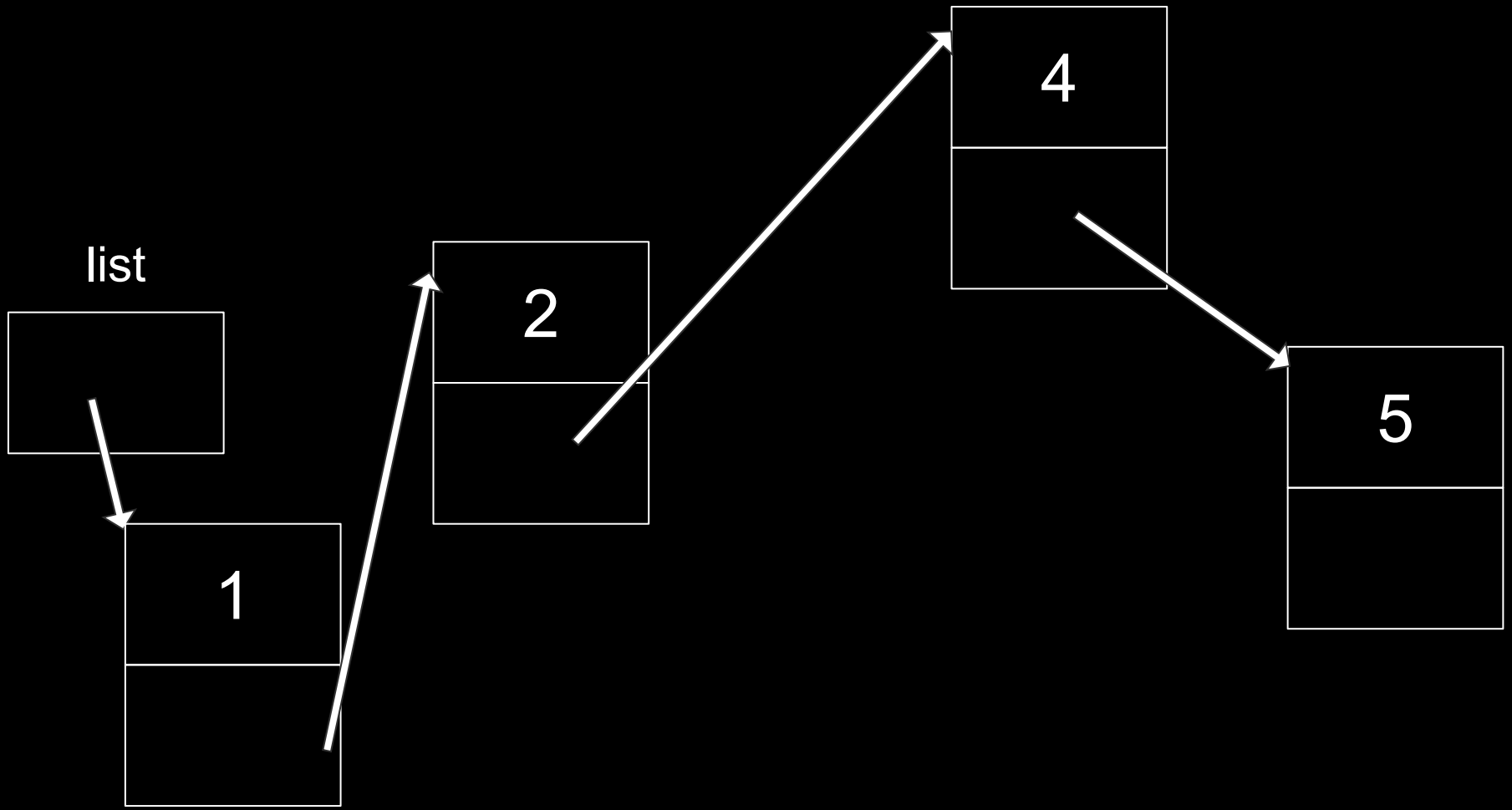

list
1
2
4
5


list
1
2
4
5

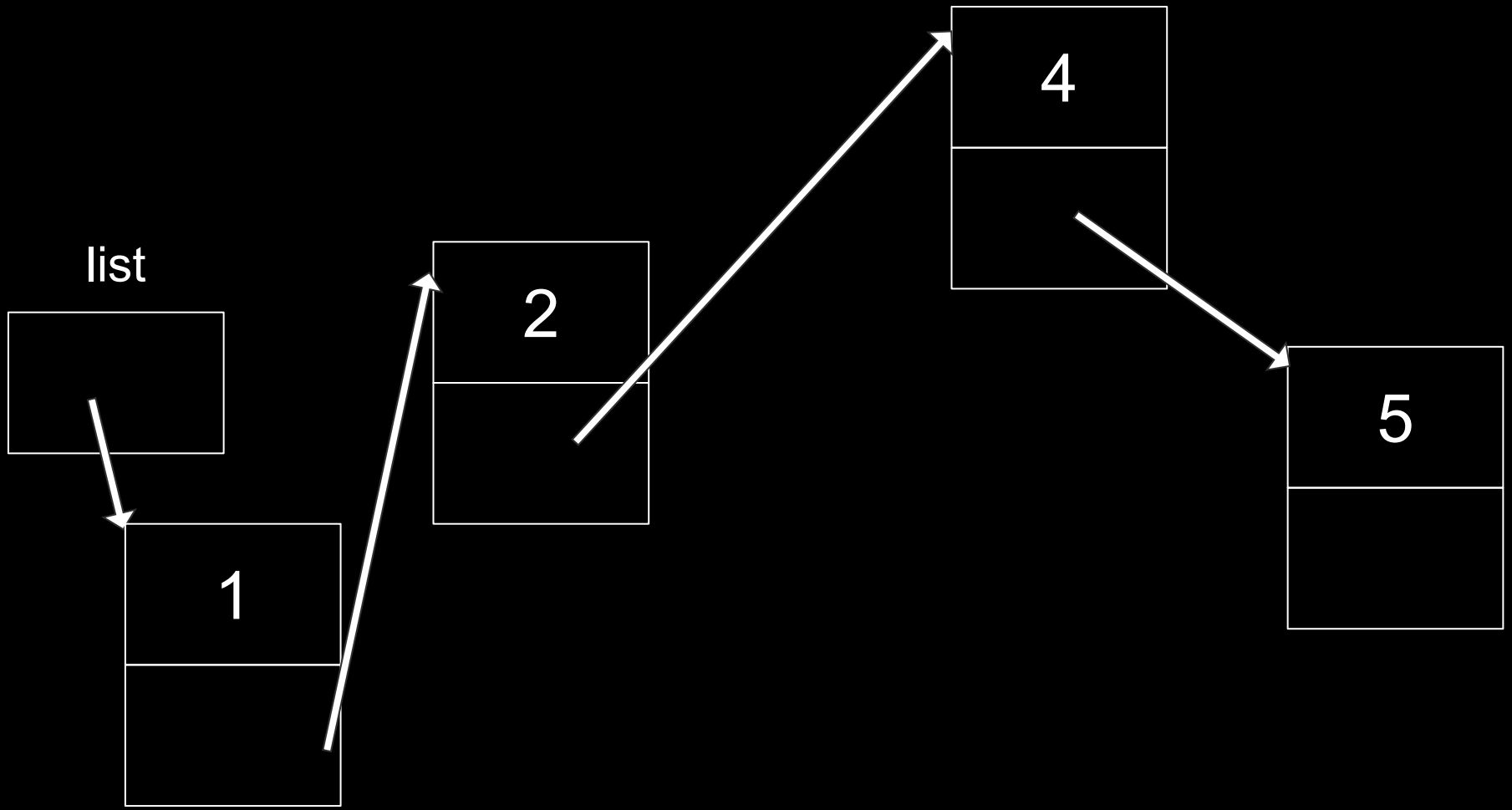

list
1
2
4
5

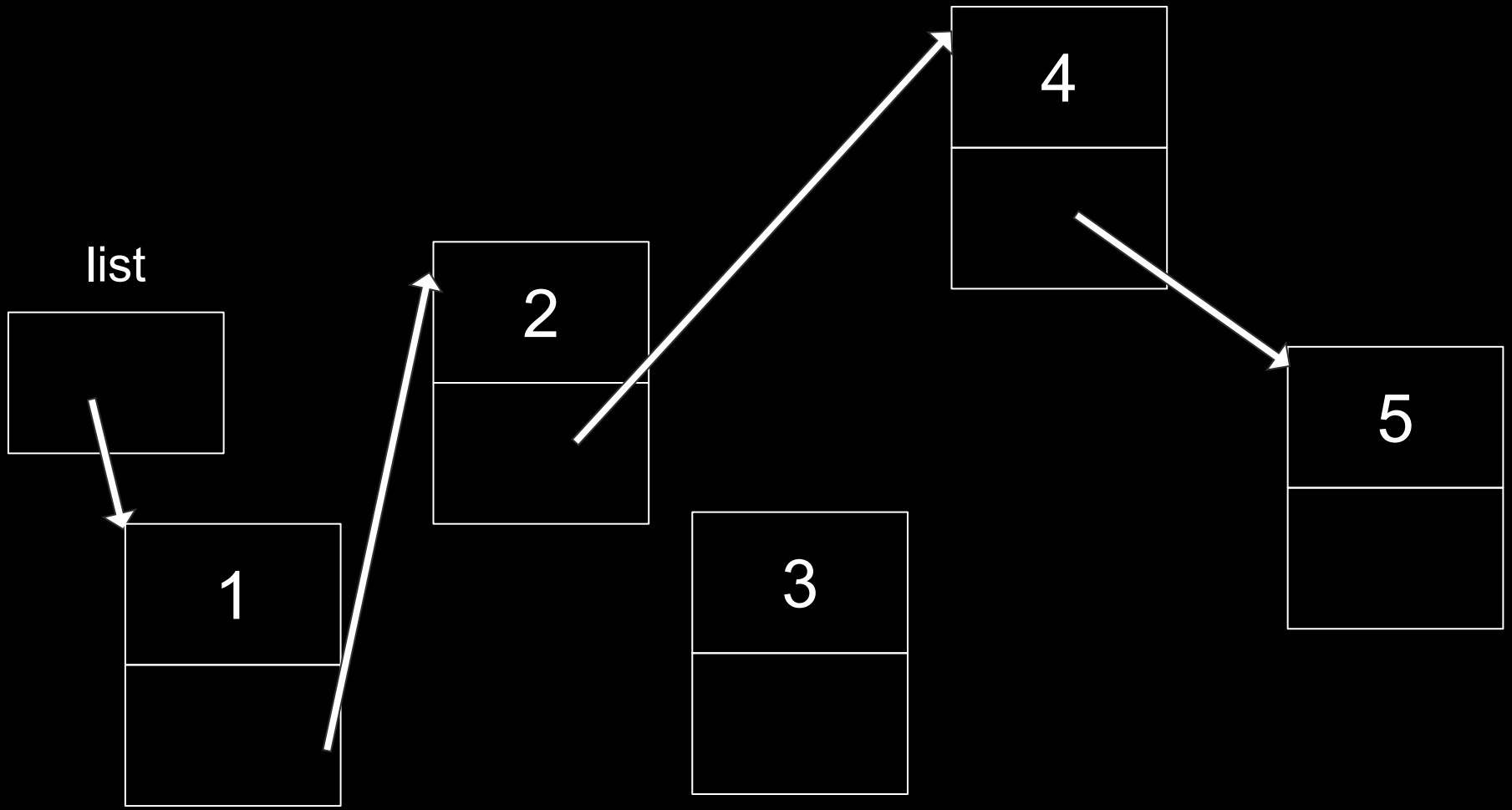

list
1
2
4
5

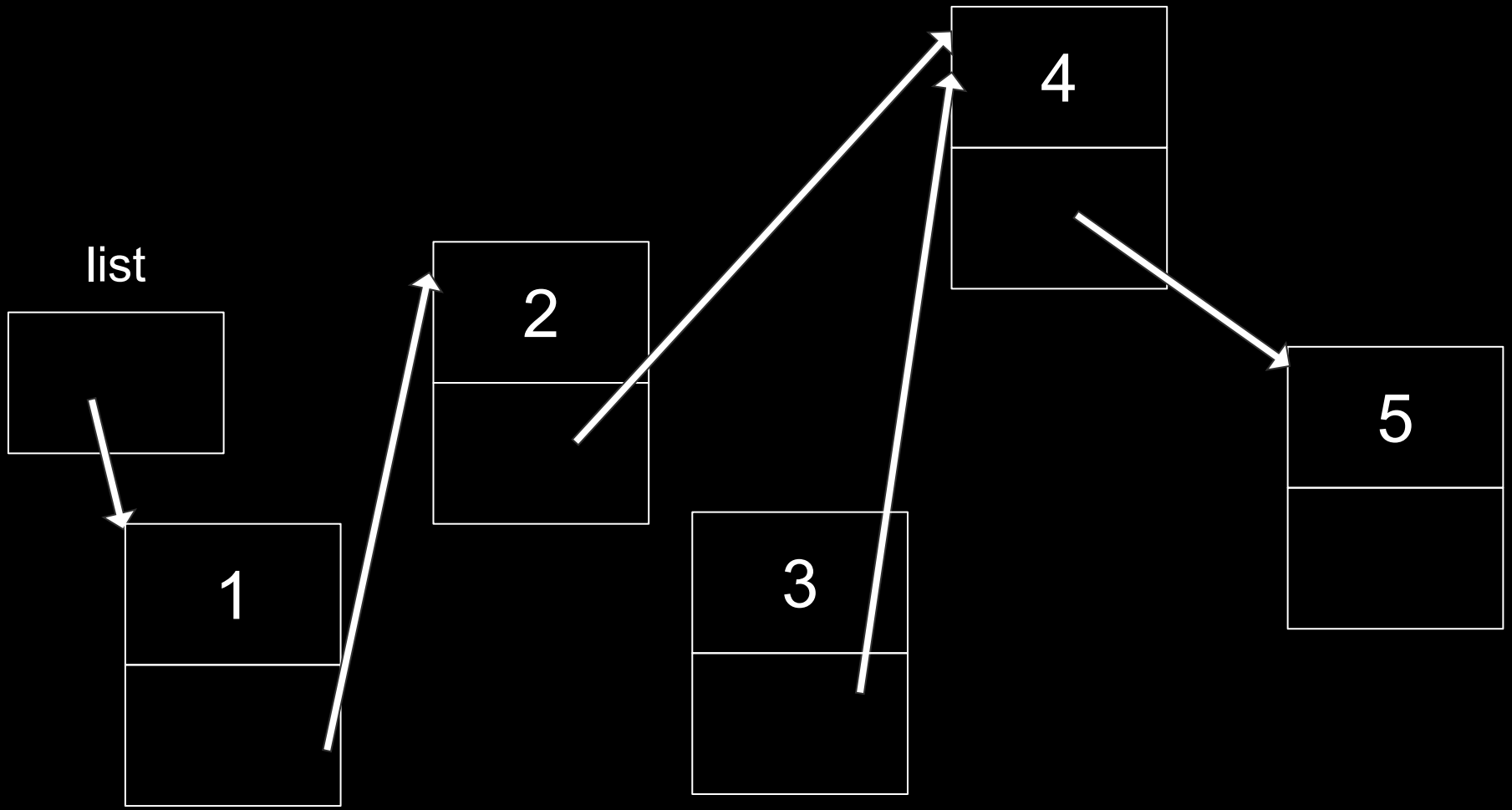

list
1
2
4
5

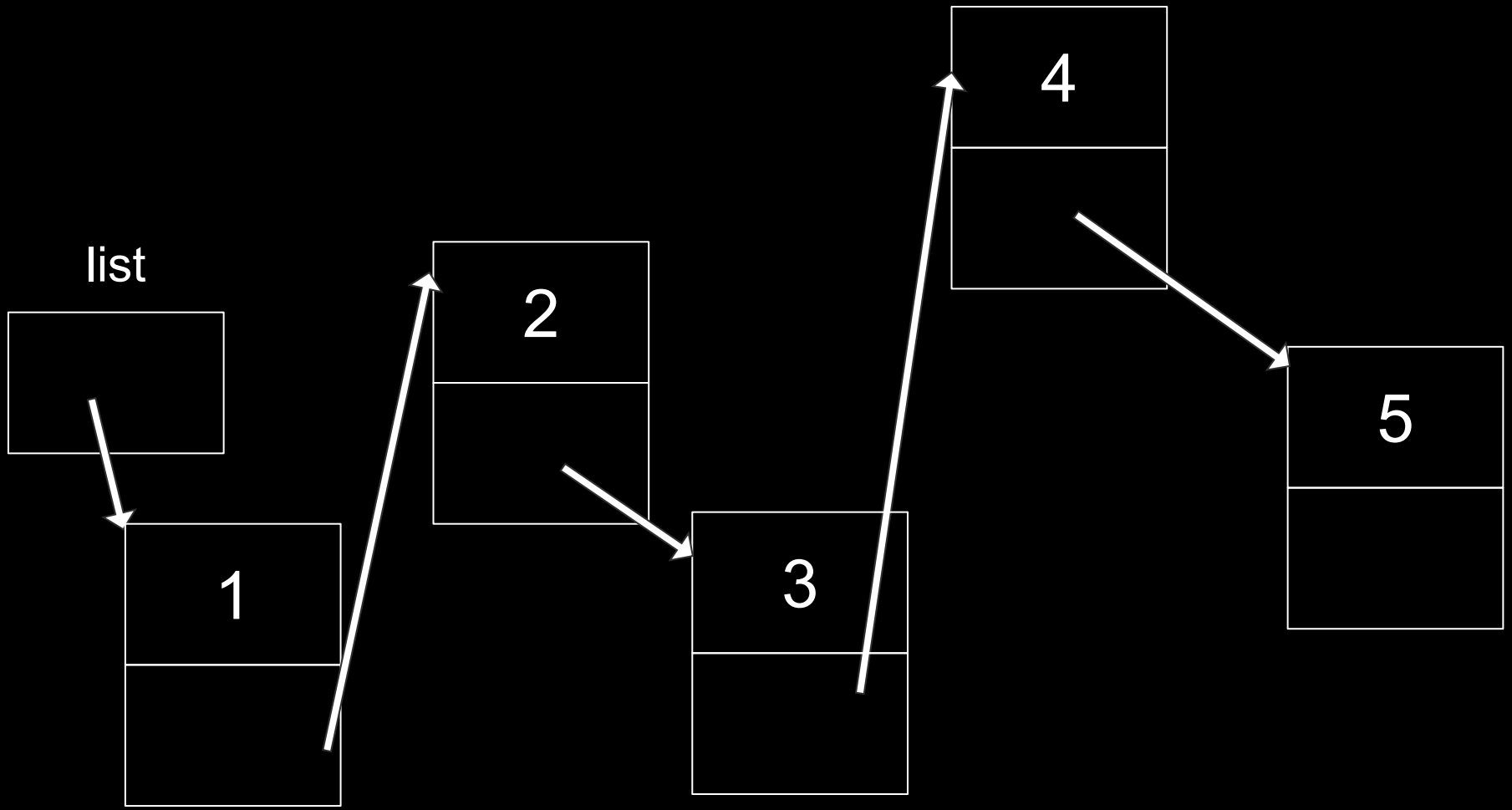

list
1
2
4
5

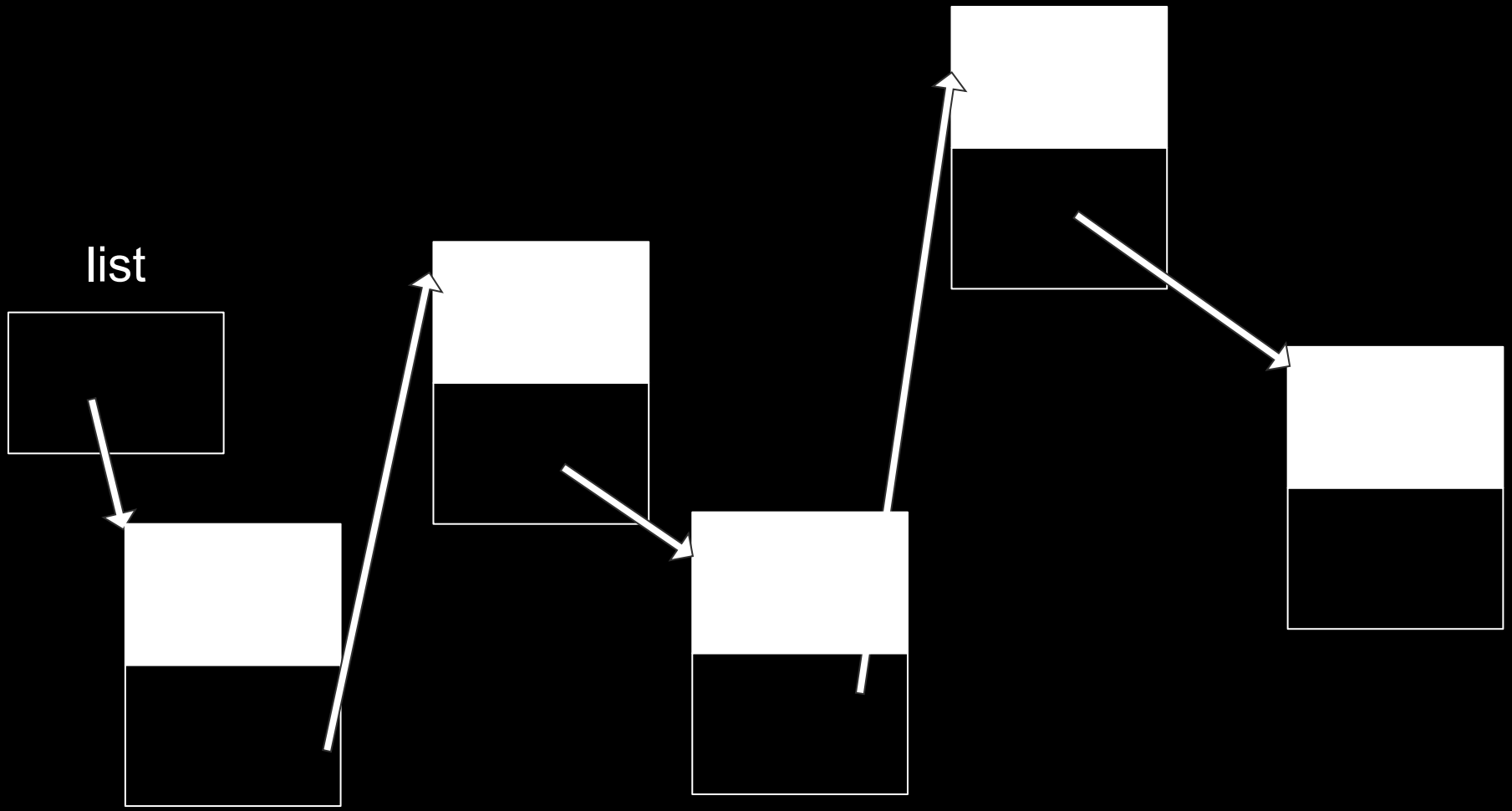
```
n.next = list;
list = n;
```

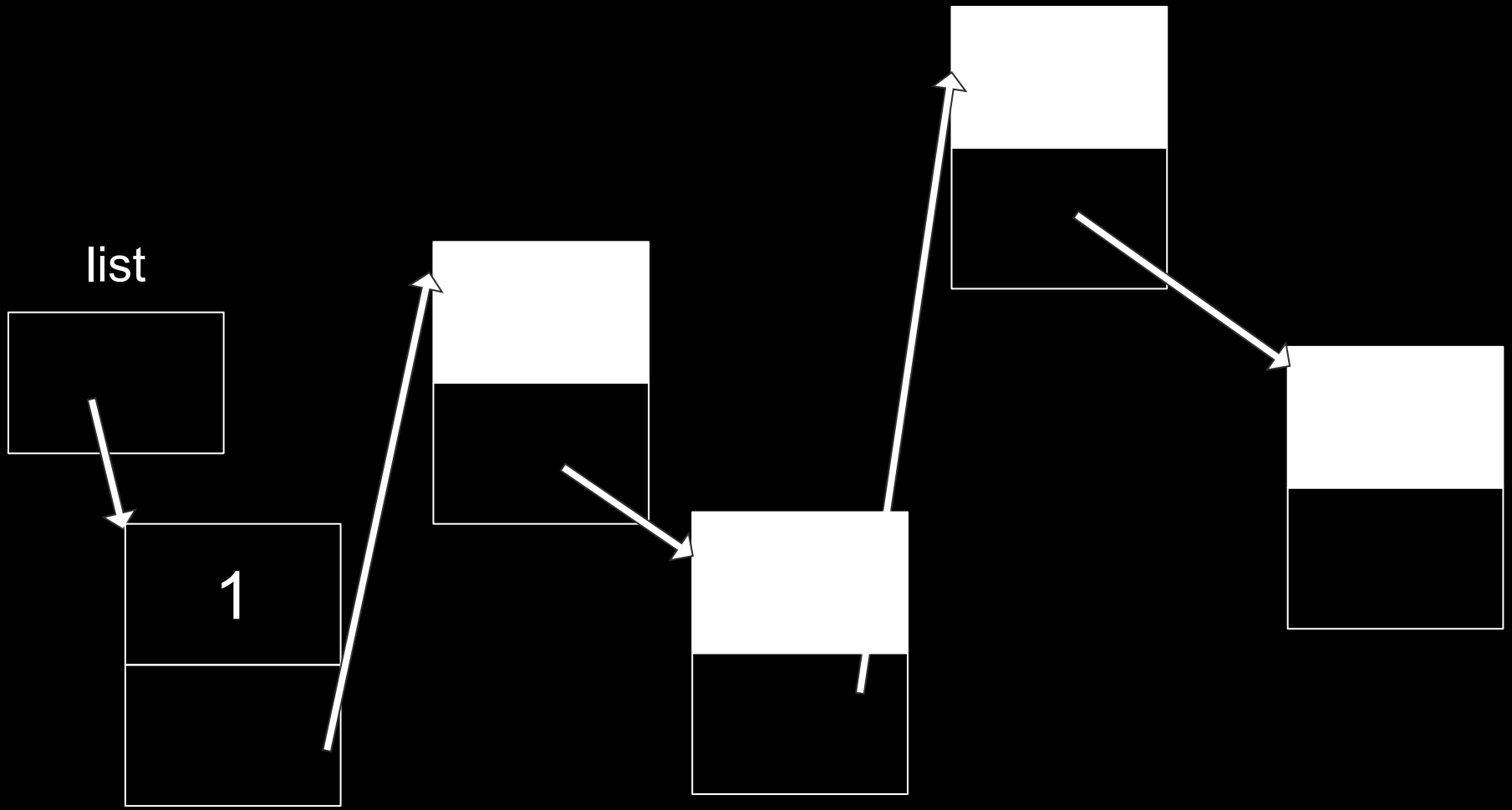

list
1
2
4
5

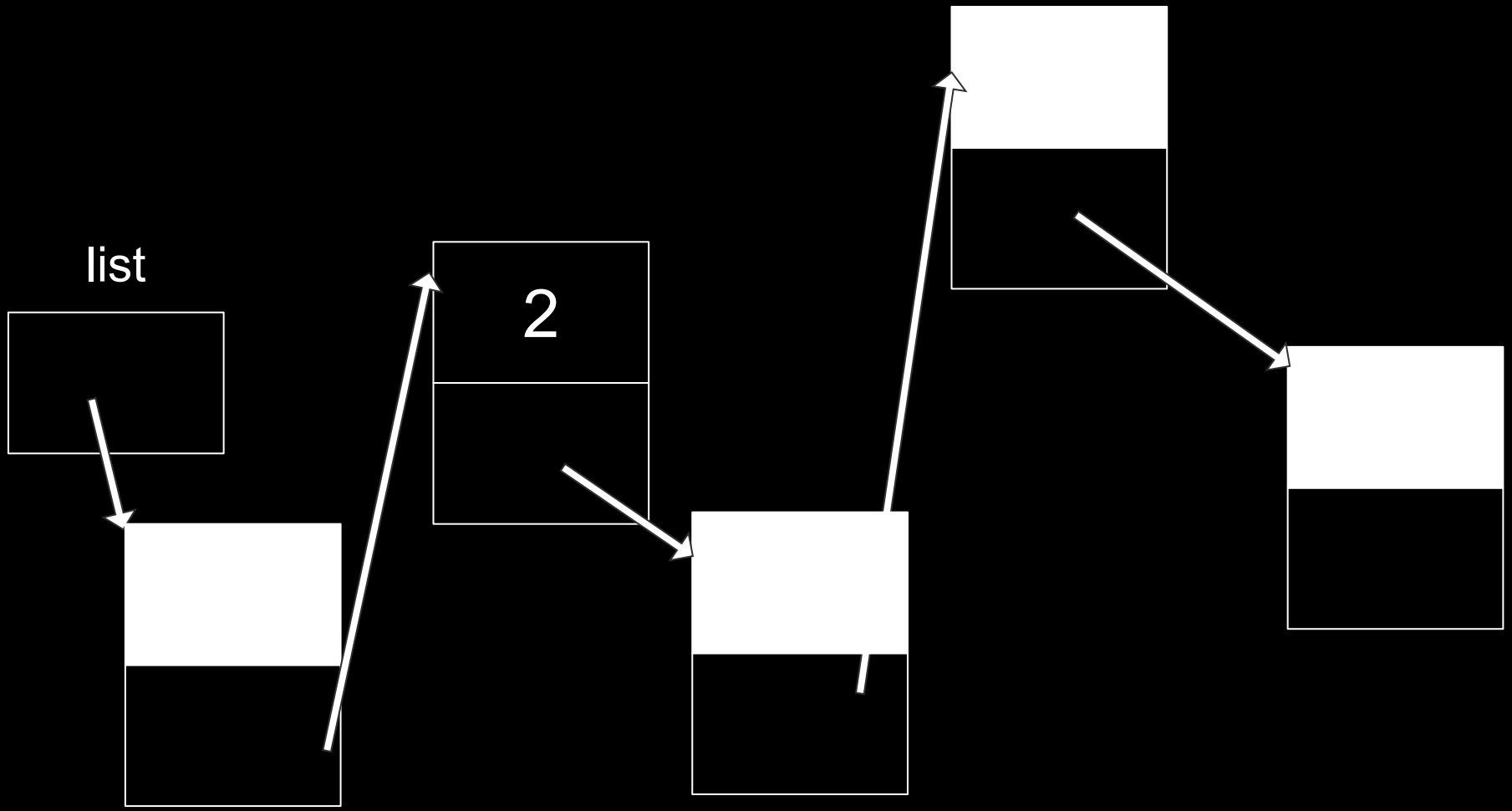

list
1
2
3
4
5

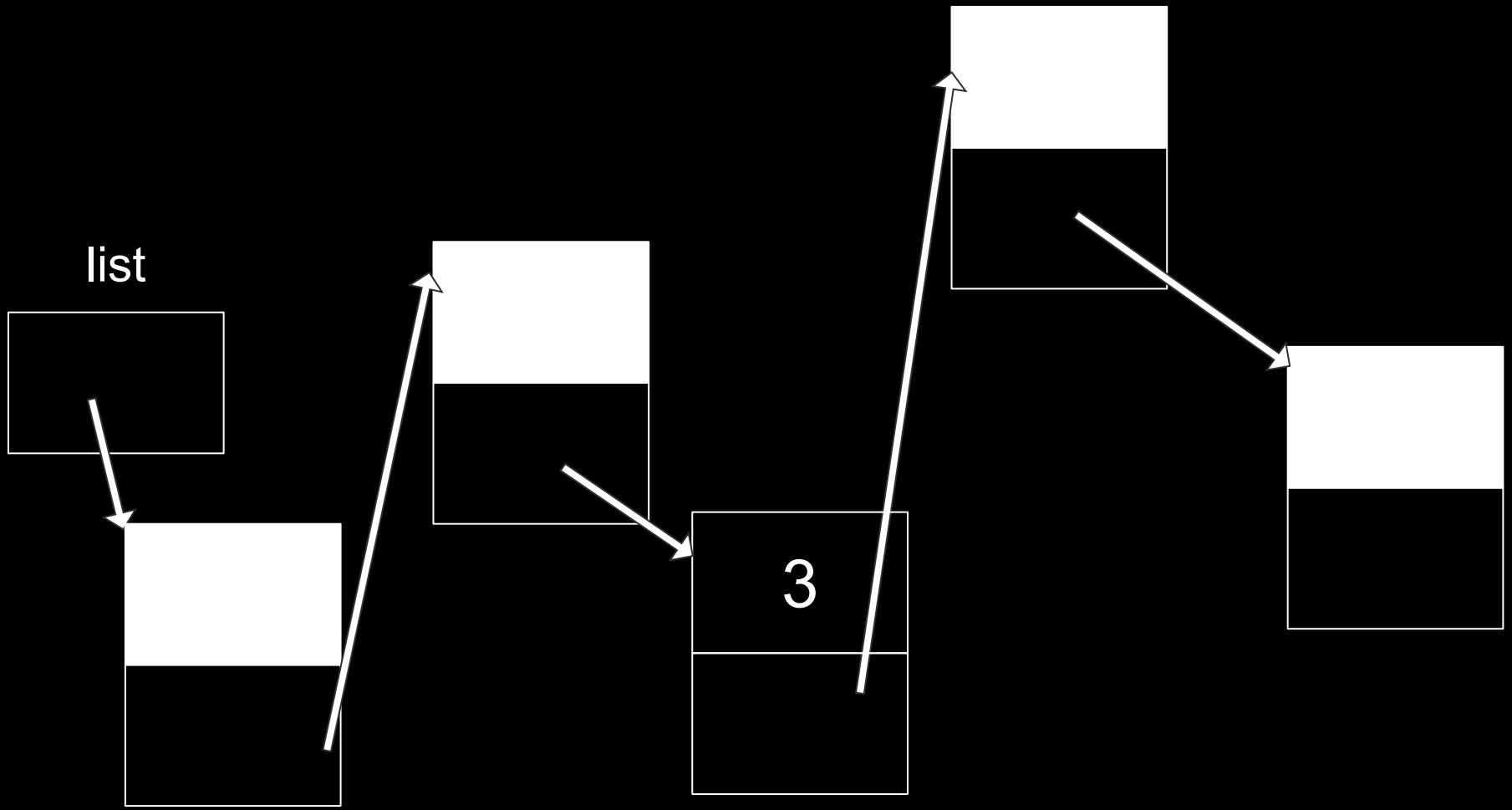

list
1
2
3
4
5

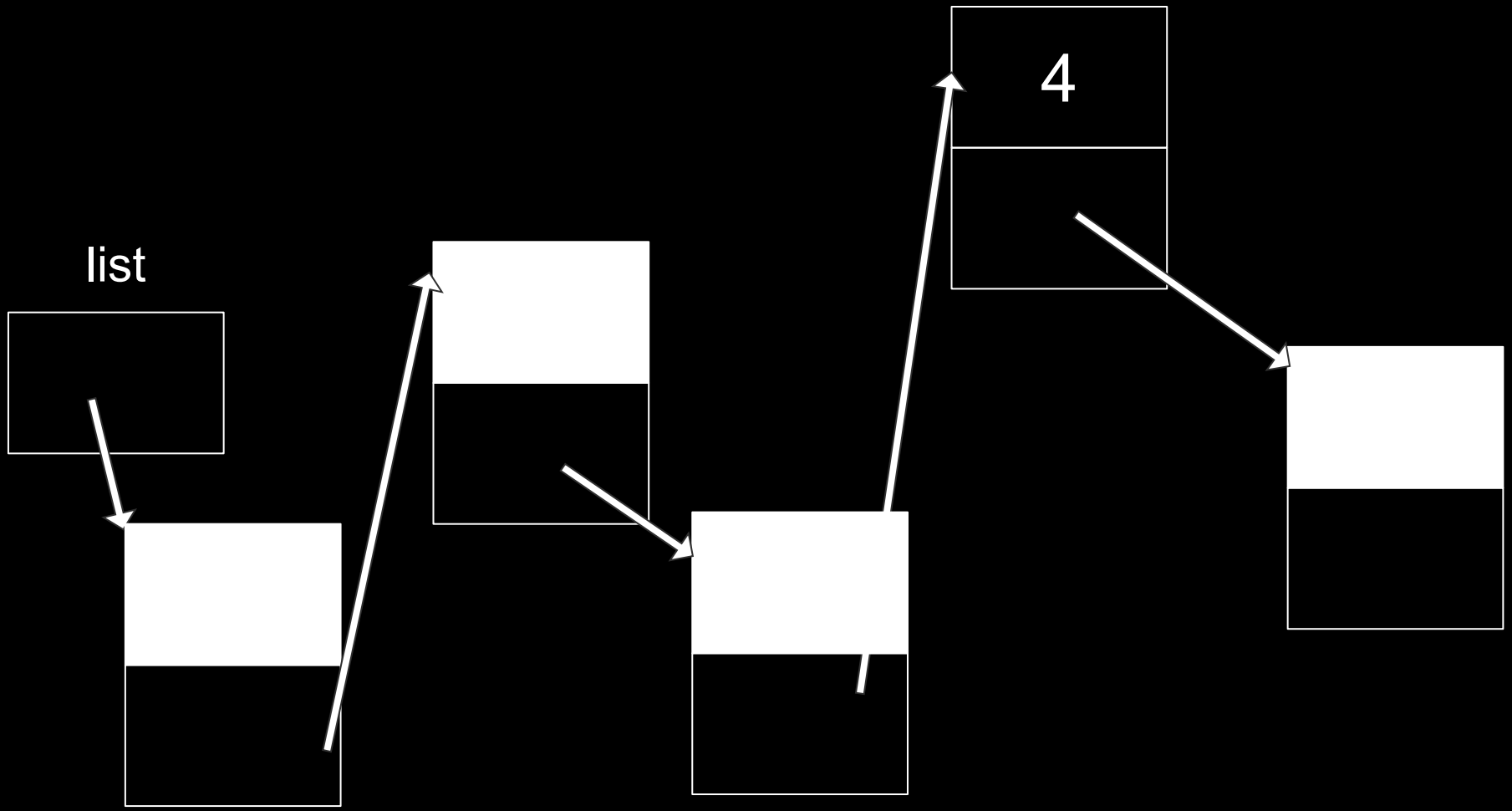

list
1
2
3
4
5

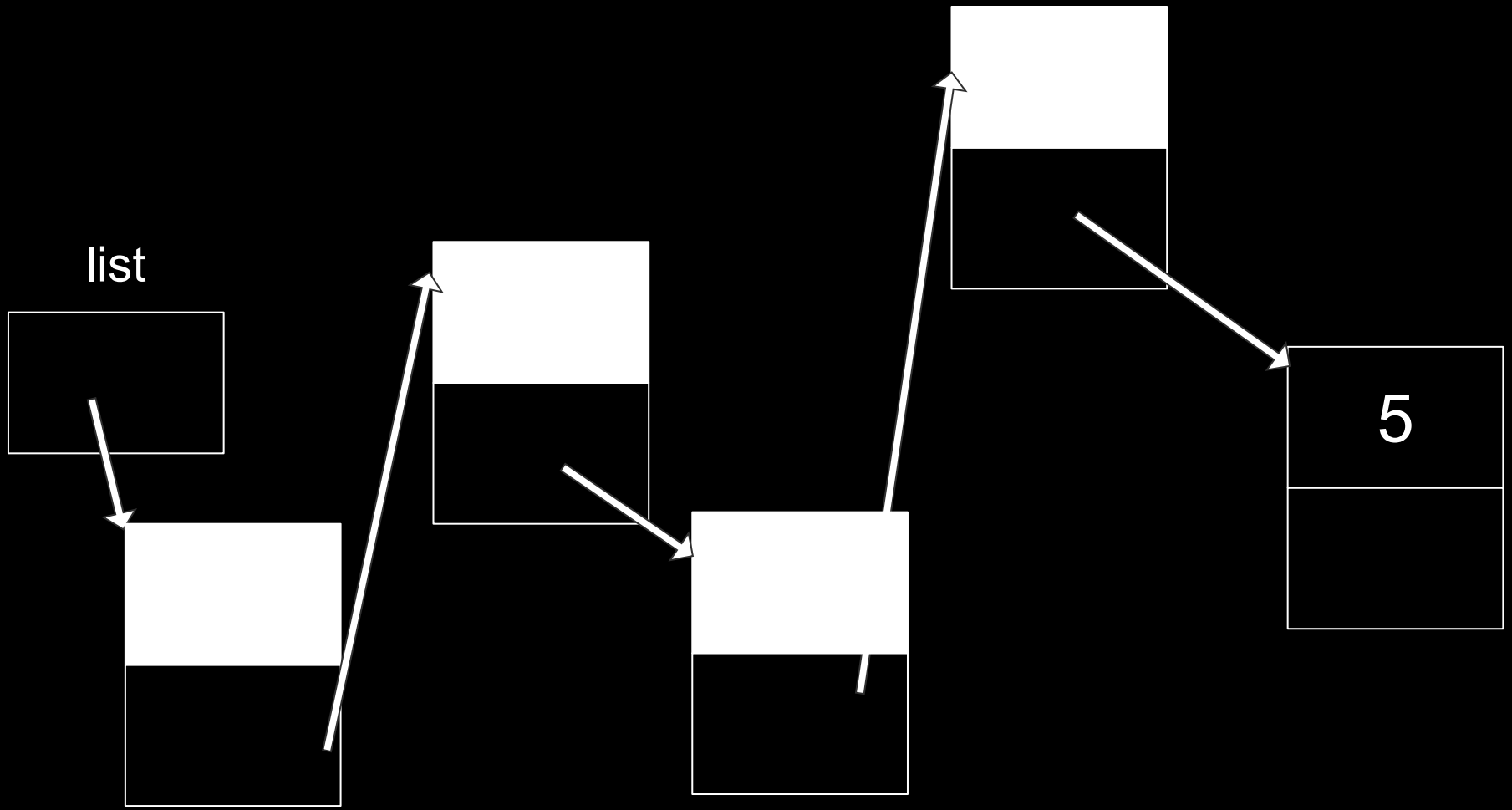

list

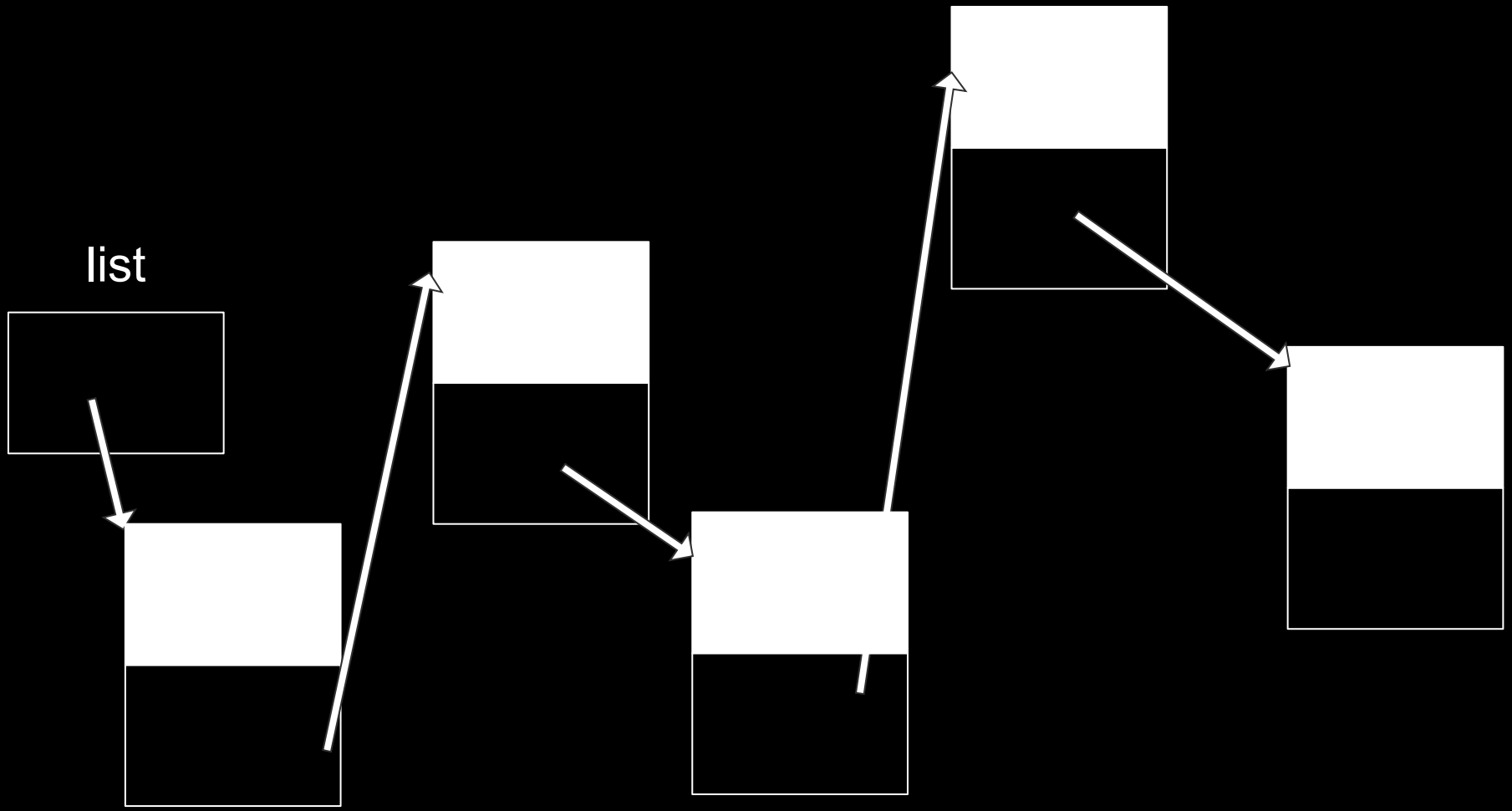

list
1


list
2


list
3


list
4


list
5


list

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

$O(n^2)$

$O(n \log n)$

$O(n)$ search

$O(\log n)$

$O(1)$

$O(n^2)$

$O(n \log n)$

$O(n)$ search, insert

$O(\log n)$

$O(1)$