Consider the following code.

class Node:
    """A node in a linked list."
    """
    def __init__(self, data, link=None):
        self.data = data
        self.link = link

    def find_max(head):
        """Return the maximum value in the linked list starting at node head."
        """
        if head is None:
            return None
        else:
            max_rest = find_max(head.link)
            if max_rest is None:
                return head.data
            else:
                return max(max_rest, head.data)

head = Node(3, Node(5, Node(1)))
m = find_max(head)

Draw pictures of the memory model just before and after each call to find_max.

* Go back to lecture notes on memory model to see how to draw pictures.

* Two important things to keep in mind:
  - EVERY value in Python is an object
  - EVERY variable name stores a _reference_ (memory address) to some object
Just before variable head is created.

```
__main__
      Node 0x01
find_max 0x02
```

Just before the first call to find_max.

```
__main__
      Node 0x01
find_max 0x02

head 0x08
      m 0x01
```

Remember:
* The line "head = Node(3, Node(5, Node(1)))" is executed like all other nested calls: inside-out.
* So Python first creates the int object with value 1, then the Node object with this int as data: when the Node constructor is executed, it simply sets the attributes of the new Node.
* After this, the Node with data = 5 (and the associated int object) are created, and then the Node with data = 3.
* Finally, variable head is set to the address of the last Node created.
* Then, Python creates variable m and calls find_max.
Just before the second call to find_max.

```
find_max
    head 0x08
    max_rest

__main__
    Node 0x01
        find_max 0x02
        head 0x08
        m
    0x03: int 1
        data 0x03
        link None
    0x04: Node
        data 0x03
        link None
    0x05: int 5
        data 0x05
        link 0x04
    0x06: Node
        data 0x07
        link 0x06
    0x07: int 3
        data 0x07
        link 0x06
```

Just before the third call to find_max.

```
find_max
    head 0x06
    max_rest

find_max
    head 0x08
    max_rest

__main__
    Node 0x01
        find_max 0x02
        head 0x08
        m
    0x03: int 1
        data 0x03
        link None
    0x04: Node
        data 0x03
        link None
    0x05: int 5
        data 0x05
        link 0x04
    0x06: Node
        data 0x07
        link 0x06
    0x07: int 3
        data 0x07
        link 0x06
```
Just before the fourth call to find_max.

At the start of the fourth call to find_max.
Just after the fourth call to `find_max`.

```
find_max
  head 0x04
max_rest None

find_max
  head 0x06
max_rest

find_max
  head 0x08
max_rest

__main__
  Node 0x01
find_max 0x02
head 0x08
m
```

```
<table>
<thead>
<tr>
<th>0x07: int</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x08: Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x05: int</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x06: Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x04: Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
```

max_rest is None and head.data has value 0x03, so...

Just after the third call to `find_max`.

```
find_max
  head 0x04
max_rest None

find_max
  head 0x06
max_rest 0x03

find_max
  head 0x08
max_rest

__main__
  Node 0x01
find_max 0x02
head 0x08
m
```

```
<table>
<thead>
<tr>
<th>0x07: int</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x08: Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x05: int</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x06: Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x04: Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>0x03: int</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x01: class</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x02: function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
```
max_rest is 0x03 and head.data is 0x05: the maximum is 0x05...
Just after the second call to find_max.

find_max
    head 0x08
    max_rest 0x05

__main__
    Node 0x01
    find_max 0x02
    head 0x08
    m

0x07: int 3
0x08: Node
data 0x07
link 0x06

0x05: int 5
0x06: Node
data 0x05
link 0x04

0x03: int 1
0x04: Node
data 0x03
link None

0x01: class ...
0x02: function ...

max_rest is 0x05 and head.data is 0x07: the maximum is 0x05...
Just after the first call to find_max.

find_max
    head 0x08
    max_rest 0x05

__main__
    Node 0x01
    find_max 0x02
    head 0x08
    m

0x07: int 3
0x08: Node
data 0x07
link 0x06

0x05: int 5
0x06: Node
data 0x05
link 0x04

0x03: int 1
0x04: Node
data 0x03
link None

0x01: class ...
0x02: function ...