

LINKED LISTS

COMPUTER SCIENCE MENTORS 61A

February 22 to February 26, 2016

For each of the following problems, assume linked lists are defined as follows:

```
class Link:

    empty = ()

    def __init__(self, first, rest=empty):
        assert rest is Link.empty or isinstance(rest, Link)
        self.first = first
        self.rest = rest
```

To check if a Link is empty, compare it against the class attribute `Link.empty`:

```
if link is Link.empty:
    print('This linked list is empty!')
```

1 What Would Python Print?

1. What will Python output? Draw box-and-pointer diagrams to help determine this.

```
>>> a = Link(1, Link(2, Link(3)))
```

```
>>> a.first
```

```
>>> a.first = 5
```

```
>>> a.first
```

```
>>> a.rest.first
```

```
>>> a.rest.rest.rest.rest.first
```

```
>>> a.rest.rest.rest = a
```

```
>>> a.rest.rest.rest.rest.first
```

2 Code Writing Questions

2. Write a function `skip`, which takes in a `Link` and returns a new `Link`.

```
def skip(l):  
    """  
    >>> a = link(1, link(2, link(3, link(4, empty))))  
    >>> link_to_list(a)  
    [1, 2, 3, 4]  
    >>> b = skip(a)  
    >>> link_to_list(b)  
    [1, 3]  
    >>> link_to_list(a)  
    [1, 2, 3, 4]  
    """
```

3. Now write function `skip` by mutating the original list, instead of returning a new list. Do NOT call the `Link` constructor.

```
def skip(lst):  
    """  
    >>> a = Link(1, Link(2, Link(3, Link(4))))  
    >>> b = skip(a)  
    >>> b  
    Link(1, Link(3))  
    >>> a  
    Link(1, Link(3))  
    """
```