Announcements

- Ants due tomorrow (tonight for +1 EC)
- HW 6 due tomorrow
- Guerrilla section this saturday 12-2PM, Soda 271/273
- MT 2 is one week from today!

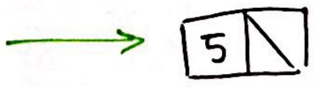
LAB 7: LINKED LISTS & TREES

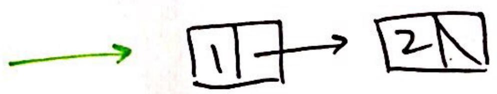
Linked List: A type of list that only stores two things: its first value and a reference to the rest of the list

* A linked list is a recursive object because the rest attribute of a single Link instance is another linked list!
→ This means rest MUST be either Link.empty or another Link instance

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

Examples

```
>>> a = Link(5)
```


```
>>> b = Link(1, Link(2))
```


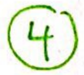
Note we define Link.second as equivalent to Link.rest.first

Tree (class implementation):

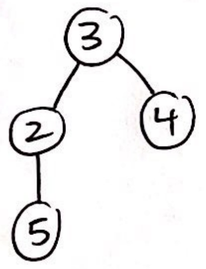
```
class Tree:
    def __init__(self, label, branches=[]):
        for b in branches:
            assert isinstance(b, Tree)
        self.label = label
        self.branches = list(branches)

    def is_leaf(self):
        return not self.branches
```

Examples

t = Tree(4) ⟶ 

t2 = Tree(3, [Tree(2, [Tree(5)]), Tree(4)])

↳ 

* Instances of the tree class are mutable!
→ You can reassign label and branches attributes