

Recursion Exercises

Exercise 1: find_list_max

- Write a recursive function called `find_list_max` that returns the maximum value in a list
- Examples:


```
>>> find_list_max([4, 13, 21, 5, 2])
21
>>> find_list_max([1, -3, 8, -5, 12])
12
```

Exercise 1: find_list_max

```
def find_list_max(t):
    """ input: a NONEMPTY list, t
        output: t's maximum element
    """
    if _____:
        _____
        base case
    elif _____:
        _____
        another case...
    else:
        _____
        another case...
```

find_list_max

```
def find_list_max(t):
    """ input: a NONEMPTY list, t
        output: t's maximum element
    """
    if len(t) == 1:
        return t[0]
    elif t[0] < t[1]: # t[0] can't be the max, remove it
        return find_list_max(t[1:])
    else: # t[1] can't be the max, remove it
        return find_list_max(t[0:1] + t[2:])
```



Try it out ...

```
Python 3.5.3 Shell
Python 3.5.1 (basepython 4.1.9 (64-bit)) (default, Jun 15 2016, 15:32:45)
Type "copyright", "credits()" or "license()" for more information.
>>>
def find_list_max(t):
    """ input: a NONEMPTY list, t
        output: t's maximum element
    """
    if len(t) == 1:
        return t[0]
    elif t[0] < t[1]: # t[0] can't be the max, remove it
        return find_list_max(t[1:])
    else: # t[1] can't be the max, remove it
        return find_list_max(t[0:1] + t[2:])
>>> find_list_max([4, 13, 21, 5, 2])
21
>>> find_list_max([1, -3, 8, -5, 12])
12
>>>
```

Exercise 2: extract a sub-list

- Write a recursive function called `extract_list` that returns a sub-list for a given range of index
- Examples:


```
>>> extract_list([4, 13, 21, 5, 2], 2, 5)
[21, 5, 2]
>>> extract_list(['hello', 'world', 'how', 'are', 'you',
'?'], 2, 5)
['how', 'are', 'you']
```

extract_list

```
def extract_list(t, low, hi):
    """ input: list t, two ints, low and hi
        output: list from low up to, not
            including hi

    """
    if hi <= low: # base case
        return []
    else:
        return
        return
```



extract_list

```
def extract_list(t, low, hi):
    """ input: list t, two ints, low and hi
        output: list from low up to, not
            including hi

    """
    if hi <= low:
        return []
    else:
        return [t[low]] + extract_list(t, low+1, hi)
```



Try it out ...

The screenshot shows a Python IDE with two panes. The left pane contains the code for the `extract_list` function. The right pane shows the output of the function when called with `extract_list(['hello', 'world', 'user', 'ace', 'you', '?'], 2, 5)`, which returns `['user', 'ace', 'you']`.

Recursion workshop

- Work on the problems in the workshop sheet in a group of two or three
- Show me the result when you complete a problem