

ICS Fall 2021

Assignment 3

Please note: For some questions, starting codes are available. Please fill the blanks in starting codes. For questions without starting codes, you can write the solutions in any form you like. Since we use an autograder to grade your code, please strictly name the functions and the submitted files as required by the questions. Moreover, when you submit your answers, you should zip the files, not the whole folder of your answers. Lastly, the starting codes may contain some tests for you to check if your code is correct.

In Python, a function can be an argument to another function. The following code snippet takes Python's built-in function `print` and passed it as an argument:

```
>>> def my_print(p, a_str):  
    p('hello {}'.format(a_str))  
  
>>> my_print(print, "ICS")  
hello ICS
```

Q.1 Write a function `order(p, q, n)` that takes two multi-element tuples `p` and `q`, and compare the element at index `n`, returns True if `p[n] > q[n]`. Examples:

```
>>> order((1, 2, 3), (2, 1, 4), 0)  
False  
>>> order((1, 2, 3), (2, 1, 4), 1)  
True  
>>> order((1, 2, 3), (2, 1, 4), 2)  
False
```

Q.2 Use `order` to implement a function `first_max(order_f, l, n)`. `l` is a list of tuples, and the function returns the **first largest** tuple in `l`. The comparison of order is done using `order` you implemented in B.1 (i.e. a tuple `p` is larger than `q` if `p[n] > q[n]`).

Example -- ('B', 6), ('X', 6) and ('P', 6) are all the largest on index 1, but ('B', 6) is the correct result because it is the *first largest*:

```
>>> tuplst = [('X', 5), ('B', 6), ('P', 4), ('X', 3), ('B', 5), ('P', 6)]
>>> print(first_max(order, tuplst, 1))
('B', 6)
```

Q.3 Implement a function `last_max(order_f, l, n)`, this time returns the *last* one.

Constraints: 1) you can *not* invoke `first_max` on a reversed list, 2) you ***must not*** modify the order function, and you can ***only*** use `order_f` to test order of two tuples.

Example: this time ('P', 6) is the *last largest*

```
>>> tuplst = [('X', 5), ('B', 6), ('P', 4), ('X', 3), ('B', 5), ('P', 6)]
>>> print(last_max(order, tuplst, 1))
('P', 6)
```