UNOFFICIAL QUIZ *for* PRACTICE SOLUTIONS

QUIZ 10

Computer Science 61A . November 12, 2015 . alvinwan.com/cs61a

This quiz will not count towards your grade. It exists to simply gauge your understanding. You will have 5 minutes to complete this quiz. In that timespan, your goal is to complete the first question and at least attempt the second.

01. ITERATORS AND GENERATORS

Complete the following implementations of map_fib_kth, both as an iterator and as a generator. Both will give every kth element of the fibonacci sequence, but with the function f applied to each one. Both should pass the following doctest:

```
>>> fib = map_fib_kth(3, lambda x: x**2)  # or MapFibKth(3, lambda x:x**2)
>>> next(fib) # starts with 0, yields 0**2
0
>>> next(fib) # skips 1 1, yields 2**2
4
>>> next(fib) # skips 3 5, yields 8**2
64
```

```
def map_fib_kth(k, f):

    curr, prev = 1, 0

    while True:

        yield f(prev)

        for _ in range(k):

            curr, prev = prev + curr, curr
```

```
class MapFibKth: # assume __init__ initializes all variables you need

    def __next__(self):

        prev = self.prev

        for _ in range(self.k):

            self.prev, self.curr = self.curr, self.curr + self.prev

        return prev
```

02. EXSTREAM

Consider a Python implementation of `Streams`. In this version, the stream has a linked list-esque representation. The `Stream` class accepts two arguments: (1) the value at the head of the linked list and (2) a function that computes the next `Stream` "node".

```
def map_fib_kth_stream(k, f, prev=0, curr=1):
    """
    >>> s = map_fib_kth_stream(3, lambda x: x**2)
    >>> s.first
    0
    >>> s.rest.first
    4
    >>> s.rest.rest.first
    64
    """
    def compute_next():  # stuck? fill out what you know about fib first

        rprev, rcurr = prev, curr

        for _ in range(k):

            rprev, rcurr = rcurr, rcurr + rprev

        return map_fib_kth_stream(k, f, rprev, rcurr)

    return Stream(f(prev), compute_next)
```