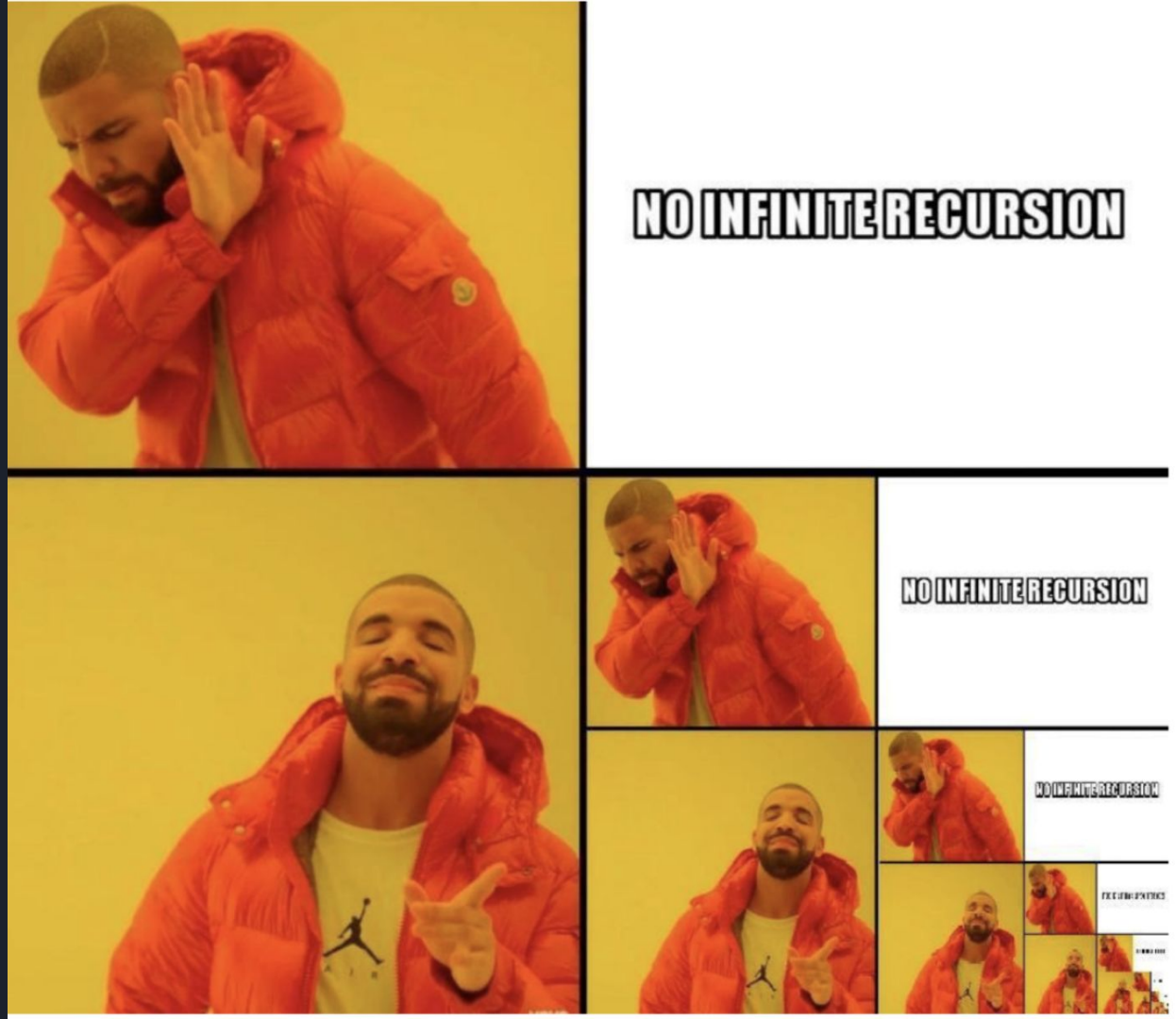
Tree Recursion

Aditya Iyer


NO INFINITE RECURSION
NO INFINITE RECURSION
NO INFINITE RECURSION

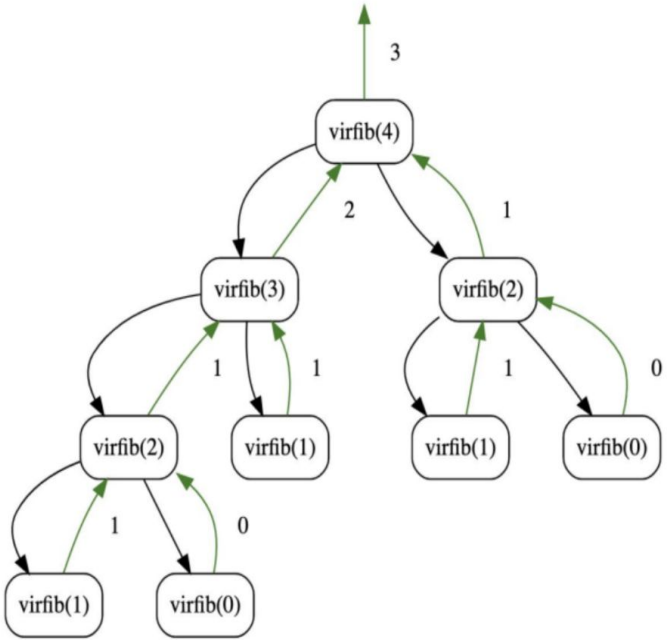
Basic Example

```
def fib(n) :
    if n == 0 :
        return 0
    elif n == 1 :
        return 1
    else:
        return fib (n - 1) + fib (n - 2)
```

Questions:

1. What is the base case?
2. What is the recursive case(s) ?
3. Draw out the tree diagram for the case when n = 4

Fibonacci Series Visualizer (n=4)



Credit: Pamela Fox, Sp2022 CS61A

Past Exam Question

4. (6.0 points) Least Resistance

Fill in the definition of the function `least_resistance`, which takes in three parameters, `m`, `n`, and `f`. `m` and `n` are integers which specify a coordinates position on a grid, and `f` is a two-argument function that takes in coordinates and returns a number. Your goal is to find the path of "least resistance" from the position `(m, n)` to the position `(0, 0)` on the grid, relative to `f`, which defines the resistance of each square, and return the total resistance met along that path.

A path is a series of consecutive steps from a coordinate position on the grid to `(0, 0)`, where at each step you may either take one step down, or one step to the left. The total resistance of a path is defined as the sum of `f` called on each coordinate position visited. For example, the below graphic visualizes the paths and of least resistance, and total resistance met, for the first two doctests.

5	25	26	29	34	41	50
4	16	17	20	25	32	41
3	9	10	13	18	25	34
2	4	5	8	13	20	29
1	1	2	5	10	17	26
0	0	1	4	9	16	25
	0	1	2	3	4	5

`f = lambda x, y: x ** 2 + y ** 2`

`50 + 41 + 32 + 25 + 18 + 13 + 8 + 5 + 2 + 1 + 0 = 195`

5	5	5	5	5	5	5
4	4	4	4	4	4	4
3	3	3	3	3	3	3
2	2	2	2	2	2	2
1	1	1	1	1	1	1
0	0	0	0	0	0	0
	0	1	2	3	4	5

`g = lambda x, y: y`

`5 + 4 + 3 + 2 + 1 + 0 + 0 + 0 + 0 + 0 + 0 = 15`

Note: In the skeleton, you are provided a line that uses `float('inf')`. This will return the Python equivalent of infinity. That is, for any number `n`, `float('inf') > n` will be `True`, no matter the value of `n`.

```
def least_resistance(m, n, f):
    """
    >>> f = lambda x, y: x ** 2 + y ** 2
    >>> least_resistance(5, 5, f)
    195
    >>> g = lambda x, y: y
    >>> least_resistance(5, 5, g)
    15
    """
    if __________:
          (a)
        return __________
                  (b)
    elif __________:
            (c)
        return float('inf')
    else:
        r1 = least_resistance(_______________________________)
                                                  (d)
        r2 = least_resistance(_______________________________)
                                                  (e)
        return __________(r1, r2) + __________
                  (f)                  (g)
```

Anonymous Feedback Form