CS 61A: Discussion 4

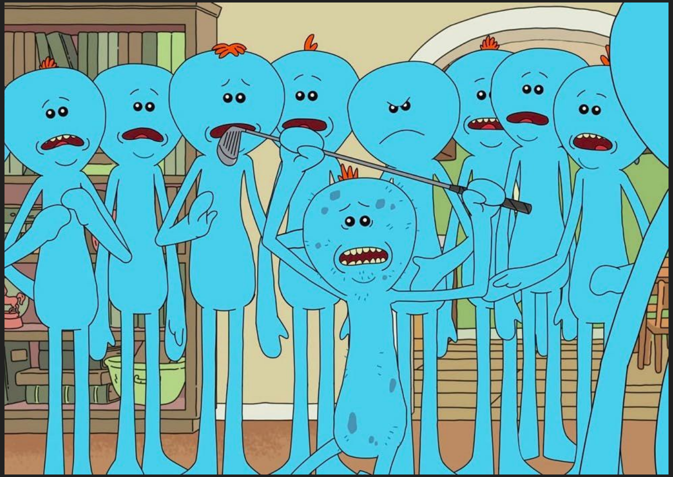
http://tinyurl.com/61a-justin-4

^do this pls

Orders of Growth

Orders of Growth

- How the runtime varies relative to the size of the input
- Big Theta notation to indicate its order without worrying about constants and smaller terms
  - reason being for a large enough input, a function of higher order than another will take longer to complete



Tips for Orders of Growth

- Nested for loops *typically* imply multiplication of runtimes (ex. Two nested for loops going through range(n) will have a runtime of n^2)
- In turn, non-nested loops typically imply addition of runtimes
    - This means the one that has a higher order will be the significant one
    - If they have the same order then total is the same order (O(x) + O(x) => O(x))
- DON'T BE FOOLED
    - Just because there is a loop that does not mean it will be linear, etc.
        - Ex. for x in range(n):
            - return 1
- Drawing diagrams especially for tree recursive problems help a lot

Worksheet Time (Section 1)

Nonlocal

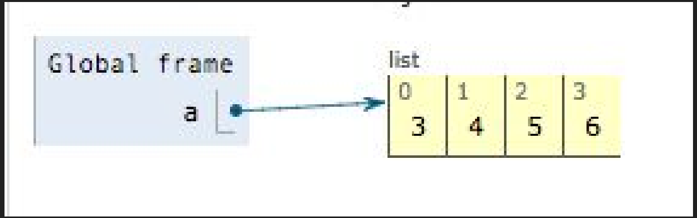
Nonlocal Things to remember

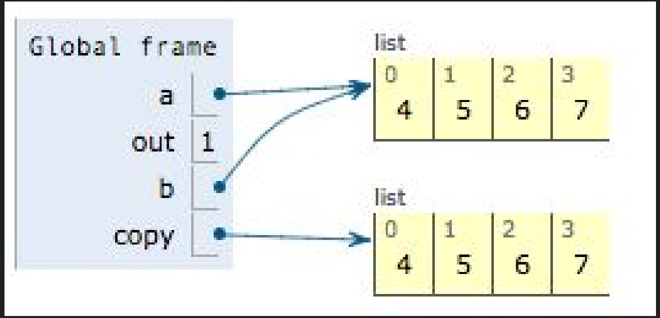
- Use it in a child frame to change the value of a variable in a parent frame (EXCLUDES GLOBAL VARIABLES)
- Errors if:
    - Nonlocal variable does not exist in parent frame(s)
    - We already have the variable defined in the current frame
- Useful in cases you have to record a running total across multiple child function calls

Box and Pointer + Mutability

List Manipulation

- Box and pointer - how we represent lists in environment diagrams!
- Can manipulate lists with append, insert, pop, remove, +
- Variables assigned to lists only carry the address of the list, so changes to list affect every variable pointing to the same list
- Better shown than explained: https://goo.gl/MPNRRW





Worksheet Time (2.1, 2.3, 3.*)