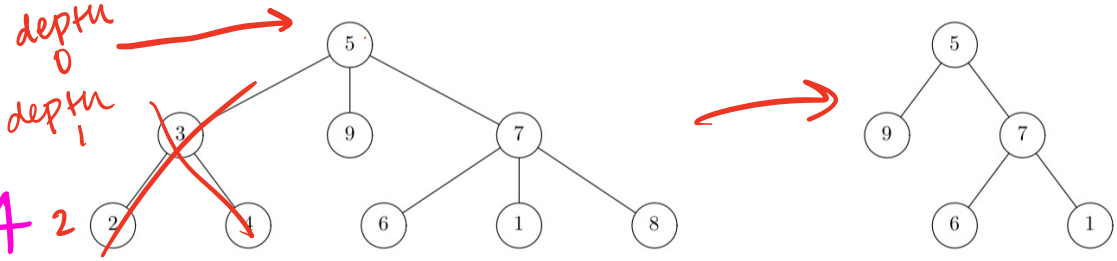
CS61A Exam Prep 14

Spring 2017 Final Q4

4. (8 points) Spring Pruning

Fill in the function **prune** below so that given a Tree (see page 2) with integer labels, it (destructively) deletes all nodes whose label is strictly less than that of their parent if their parent is at even depth, or whose label is strictly greater that that of their parent if their parent is at odd depth. Deleting a node deletes the entire subtree below it. The root of the entire tree is at depth 0. For example, given the tree on the left, your function should produce the tree on the right without creating any new Tree nodes. The function **prune** does not return a value.




```
def prune(t):
    def prune_level(t, d):
        if d % 2 == 0:
            t.branches = [b for b in t.branches if b.label >= t.label]
        else:
            t.branches = [b for b in t.branches if b.label <= t.label]
        for b in t.branches:
            prune_level(b, d+1)
    prune_level(t, 0)
```

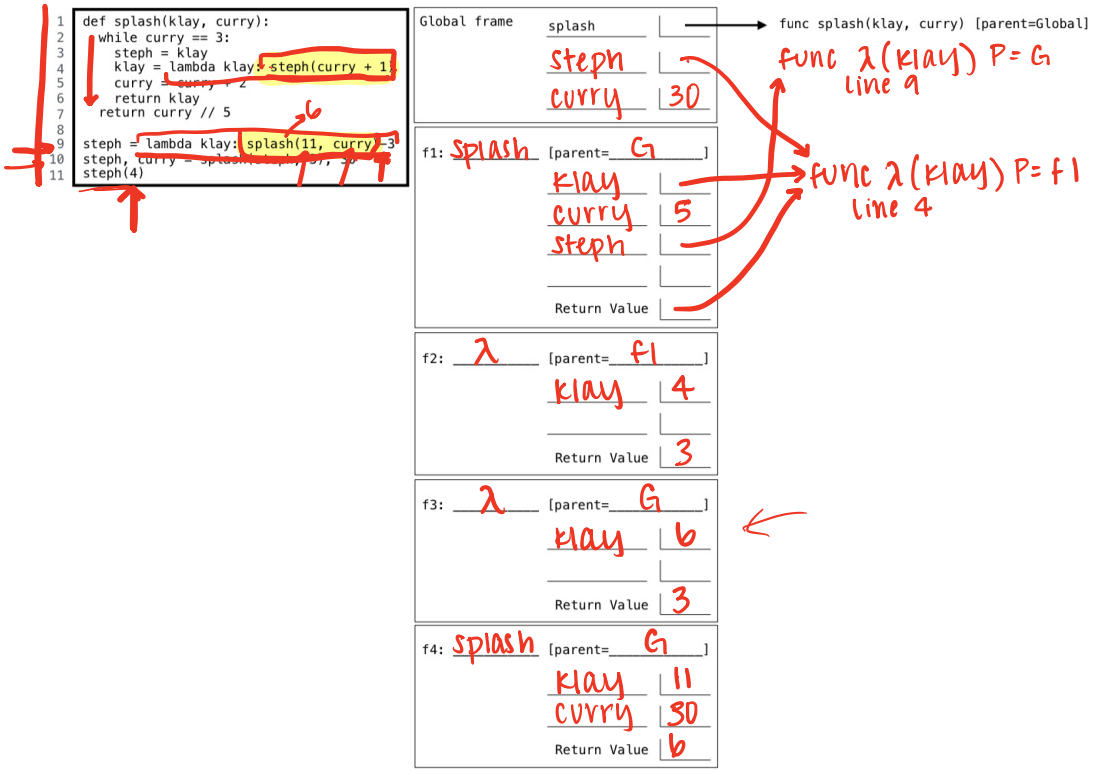
2. (8 points) Goldeen State

Fill in the environment diagram that results from executing the code below until the entire program is finished, an error occurs, or all frames are filled. *You may not need to use all of the spaces or frames.*

A complete answer will:

- Add all missing names and parent annotations to all local frames.
- Add all missing values created or referenced during execution.
- Show the return value for each local frame.

Fall 2016 Midterm 1 Q2



Summer 2015 final Q1

melaniecooray@berkeley.edu

1. (14 points) What Would Scheme Do?

Assume that you have started the interactive Scheme interpreter from Project 4 using `python3 scheme.py` and executed the following statements:

```
(define lst '(1 2 (+ 3 4)))

(define y 1)

(define f (mu (x) (+ x y)))

(define g (lambda (x y) (f (+ x x))))

(define (h z) (z 3 7))
```

For each of the expressions in the table below, write the output displayed by the interpreter when the expression is evaluated. If an error occurs, write "Error". Assume each is executed in the same interactive session started above; **the result of executing one line may affect the result of other lines.**

Expression	Interactive Output
`(+ 1 2 3)`	6
`(/ 1 0)`	Error
`lst`	`(1 2 (+ 3 4))`
`(car (cdr lst))`	2
~~`(cons (cons 3 (cons 4 nil)) 5)`~~	
`(or 1 (/ 1 0))`	1
`(and 1 (/ 1 0))`	Error
`(h +)`	
`(define x 5)`	#undefined
`(f (+ x x))`	#f
`(h g)`	
`(let ((a 1) (2 2)) a)`	
`(let ((x 1) (y x)) y)`	
`(f y)`	

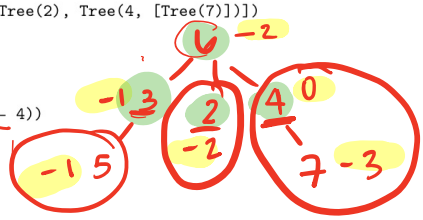
FALL 2019 Midterm 2 Q3

3. (6 points) Max Tree

(a) (4 pt) Implement `max_tree`, which takes a `Tree` instance `t` and a function `key` that takes one argument and returns a number. The `max_tree` function returns the label `n` of `t` for which `key(n)` is largest. If there is more than one label for which `key` returns the same largest value, `max_tree` can return any of those labels. The `Tree` class is defined on the Midterm 2 Study Guide. You may **not** call `min` or `max`.

```
def max_tree(t, key):
    """Return the label n of t for which key(n) returns the largest value.

    >>> t = Tree(6, [Tree(3, [Tree(5)]), Tree(2), Tree(4, [Tree(7)])])
    >>> max_tree(t, key=lambda x: x)
    7
    >>> max_tree(t, key=lambda x: -x)
    2
    >>> max_tree(t, key=lambda x: -abs(x - 4))
    4
    """
    if t.is_leaf():
        return t.label
    x = t.label
    for b in t.branches:
        m = max_tree(b, key)
        if key(x) < key(m):
            x = m
    return x
```



label w/ largest value

label in the tree

x is a label in the tree

(b) (2 pt) Now implement `max_tree` in one line using a call to the built-in `max` function.
You may **not** include any additional calls to `max` beyond the one included in the template below.

```
def max_tree(t, key):
    "Return the label n of t for which key(n) returns the largest value."
    return max([t.label] + [max_tree(b, key) for b in t.branches], key=key)
```