

Q6: (Warmup) Height

Write a function that returns the height of a tree. Recall that the height of a tree is the length of the longest path from the root to a leaf.

```
def height(t):  
    """Return the height of a tree.  
  
    >>> t = tree(3, [tree(5, [tree(1)]), tree(2)])  
    >>> height(t)  
    2  
    """  
    "*** YOUR CODE HERE ***"
```

Q2: (Tutorial) Max Product

Write a function that takes in a list and returns the maximum product that can be formed using nonconsecutive elements of the list. The input list will contain only numbers greater than or equal to 1.

```
def max_product(s):  
    """Return the maximum product that can be formed using non-consecutive  
    elements of s.  
  
    >>> max_product([10,3,1,9,2]) # 10 * 9  
    90  
    >>> max_product([5,10,5,10,5]) # 5 * 5 * 5  
    125  
    >>> max_product([])  
    1  
    """"  
    "*** YOUR CODE HERE ***"
```

Q5: (Tutorial) Add This Many

Write a function that takes in a value `x`, a value `el`, and a list `s` and adds as many `el`'s to the end of the list as there are `x`'s. **Make sure to modify the original list using list mutation techniques.**

```
def add_this_many(x, el, s):
    """ Adds el to the end of s the number of times x occurs
    in s.

    >>> s = [1, 2, 4, 2, 1]
    >>> add_this_many(1, 5, s)
    >>> s
    [1, 2, 4, 2, 1, 5, 5]
    >>> add_this_many(2, 2, s)
    >>> s
    [1, 2, 4, 2, 1, 5, 5, 2, 2]
    """

    "*** YOUR CODE HERE ***"
```

Q8: (Tutorial) Find Path

Write a function that takes in a tree and a value `x` and returns a list containing the nodes along the path required to get from the root of the tree to a node containing `x`.

If `x` is not present in the tree, return `None`. Assume that the entries of the tree are unique.

For the following tree, `find_path(t, 5)` should return `[2, 7, 6, 5]`

```
def find_path(tree, x):  
    """  
    >>> t = tree(2, [tree(7, [tree(3), tree(6, [tree(5), tree(11)])] ), tree(15)])  
    >>> find_path(t, 5)  
    [2, 7, 6, 5]  
    >>> find_path(t, 10) # returns None  
    """  
    if _____:  
        return _____  
    _____:  
        path = _____  
        if _____:  
            return _____
```

