Quiz 3

Kevin Lin

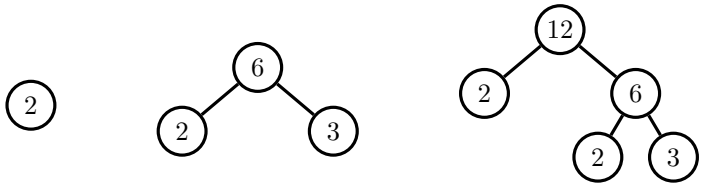
September 22, 2016

1. **Environment Diagrams**

```
def reverse(lst):
    if len(lst) <= 1:
        return lst
    return reverse(lst[1:]) + [lst[0]]

lst = [1, [2, 3], 4]
rev = reverse(lst)
```

2. **Trees** We can represent the factorization of a number with a *full binary tree*: a tree that has either two subtrees or none at all. Define `factor_tree` which takes an integer n greater than one and returns a factor tree for n.



Recall that a factor tree contains only the **prime factors of n** with the exception of the root, n, itself. The `tree` abstraction appears below.

```
def tree(root, branches=[]):
    return [root] + branches

def root(tree):
    return tree[0]

def branches(tree):
    return tree[1:]

def factor_tree(n):
```

Now, write a procedure, `count`, which counts the number of instances that a prime factor, p, appears in the factor tree t.

```
def count(t, p):
```