Stephanie Doan
ID: 604981556

Homework #1

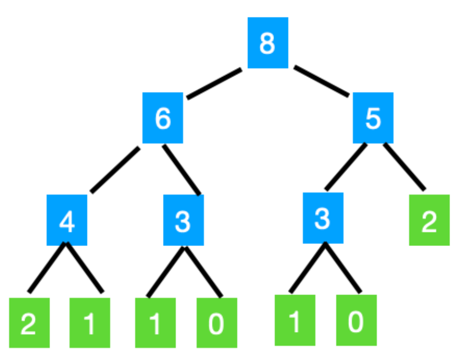
1. When larger values of N are tested with my written PAD function, the program takes a long time, often several minutes, to run the function. This is because my PAD function uses binary recursion, calling PAD(n-2) and PAD(n-3) within every function call. Each call of PAD branches into two more calls, so the algorithm grows exponentially. Therefore, increasing N severely increases the runtime of the PAD algorithm. An iterative solution would be favored for space and efficiency, in which previous values are stored to calculate the next value.

   Test cases:

```
PAD(0) = 1
PAD(1) = 1
PAD(2) = 1
PAD(3) = 2
PAD(4) = 2
PAD(5) = 3
PAD(6) = 4
PAD(7) = 5
PAD(8) = 7
PAD(9) = 9
PAD(10) = 12
```

2. The relationship between the output of SUMS(n) and PAD(n) is: *SUMS(n) = PAD(n) - 1*. This is because when constructing a full binary tree representing the recursive calls of PAD, we will have a full binary tree with *N* leaf nodes and *2N-1* total nodes. The number of leaf nodes, *N*, is the result of PAD(n), and the total number of nodes, *2N-1*, represents the total number of calls to PAD. However, the leaf nodes (N total) are PAD calls that do not perform addition since they simply return 1, while each non-leaf node performs one addition. Therefore, *(2N-1)-N = N-1*, so *SUMS(n) = PAD(n) -1*.

   For example, in the binary tree below, constructed when PAD is called on n=8, the function calls create a binary tree with 13 total nodes and 7 leaf nodes, so the *SUMS(8)=6* and *PAD(8)=7*, which fits with the observation that *SUMS(n) = PAD(n) - 1* since *6=((7*2)-1)-7*.



Test cases:

```
SUMS(0) = 0
SUMS(1) = 0
SUMS(2) = 0
SUMS(3) = 1
SUMS(4) = 1
SUMS(5) = 2
SUMS(6) = 3
SUMS(7) = 4
SUMS(8) = 6
SUMS(9) = 8
SUMS(10) = 11
```

3. Test cases:

```
(ANON ’42)                                  >      ?
(ANON 'FOO)                                 >      ?
(ANON '(((L E) F) T))                       >      (((? ?) ?) ?)
(ANON '(5 FOO 3.1 -0.2))                    >      (? ? ? ?)
(ANON '(1 (FOO 3.1) -0.2))                  >      (? (? ?) ?)
(ANON '(((1 2) (FOO 3.1)) (BAR -0.2)))      >      (((? ?) (? ?)) (? ?))
(ANON '(R (I (G (H T)))))                   >      (? (? (? (? ?))))
```