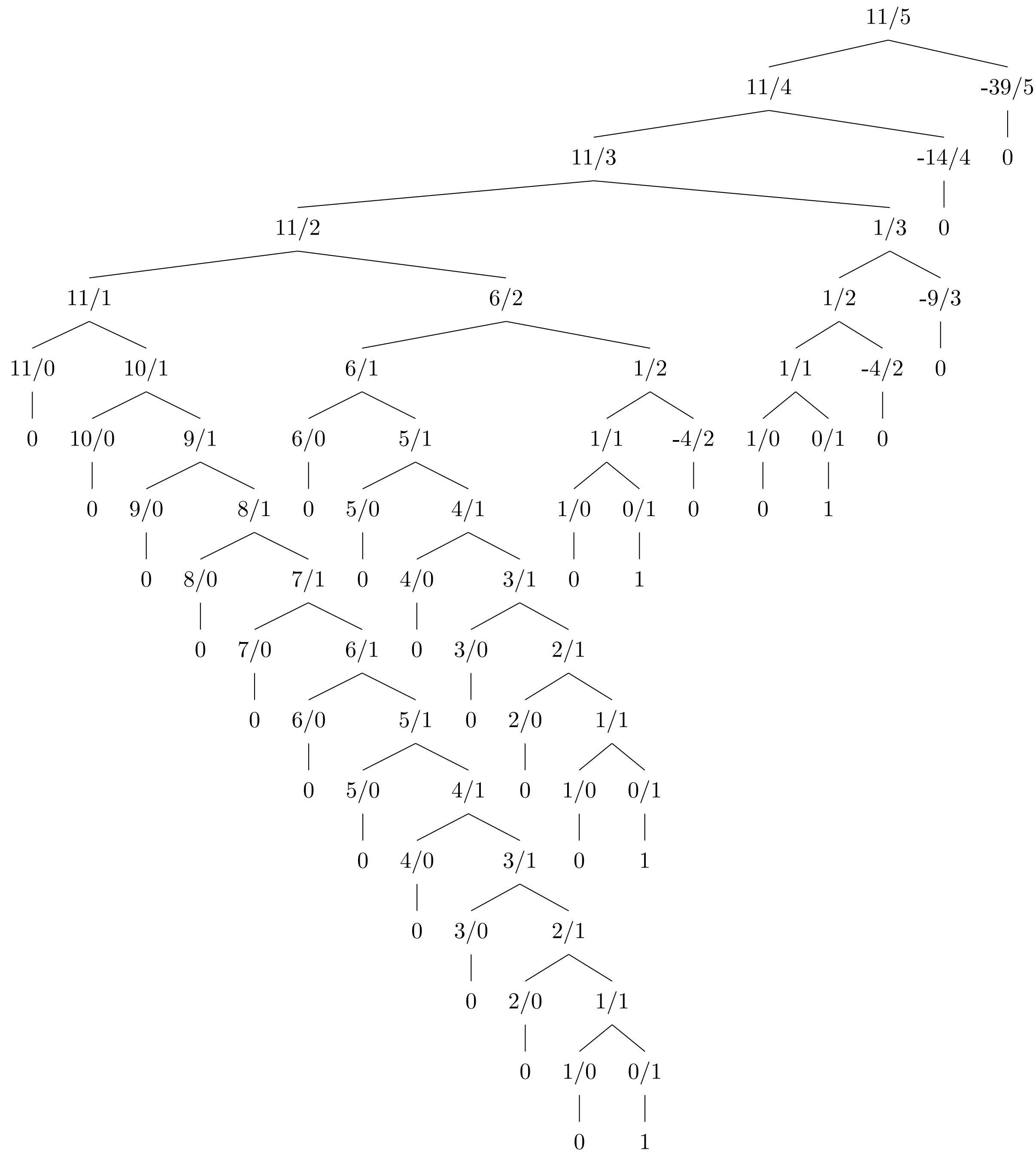
Let i/j be a call to (*cc* i j). The following is the recursion tree generated for the *count-change* procedure when making change for 11 cents.

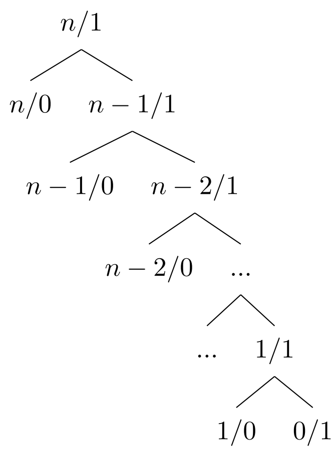


1 Space Complexity

The space required for the program is proportional to the maximum height of the recursion tree at any point in the procedure. The height of the recursion tree will be dominated by the successive calls to cc with the amount decreasing by 1 each time. Since this is proportional to n, the space complexity of the procedure is $\Theta(n)$.

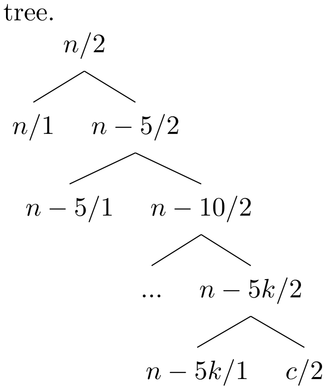
2 Time Complexity

Let's consider the case where $k = 1$. Then we have the following recursion tree.



Then, $T(n, 1) = 2n + 1 = \Theta(n)$.

Now let's consider the case where $k = 2$. We now have the following recursion tree.



where $c \leq 0$. Then, $T(n, 1) = \lfloor n/2 \rfloor (2n + 2) + 2 = \Theta(n^2)$.

In general, for $k \geq 0$, $T(n, k) = \Theta(n^k)$. Since $k = 5$ for the coin change procedure, the time complexity will be (n^5).