```
    return c
```

The integer division is exact at every step by that identity. The executable study compares all three, including the two functions specifically requested by the source. It uses separate `timeit.Timer.repeat` measurements, records repetitions and verifies equality before timing.

Comparative cost and preferred method

Counting each recursive invocation gives $T_0 = 1$ and $T_n = 1 + 2\sum_{i<n} T_i$, hence $T_n = 3^n$. The dynamic algorithm performs $n(n+1)/2$ products and stores $n+1$ integers. The multiplicative algorithm performs n updates and stores one Catalan value. These are exact counts in the indicated models. Timing tables report actual measurements separately and include the recursive setup; integer bit growth is retained as a limitation of unit-cost counting.