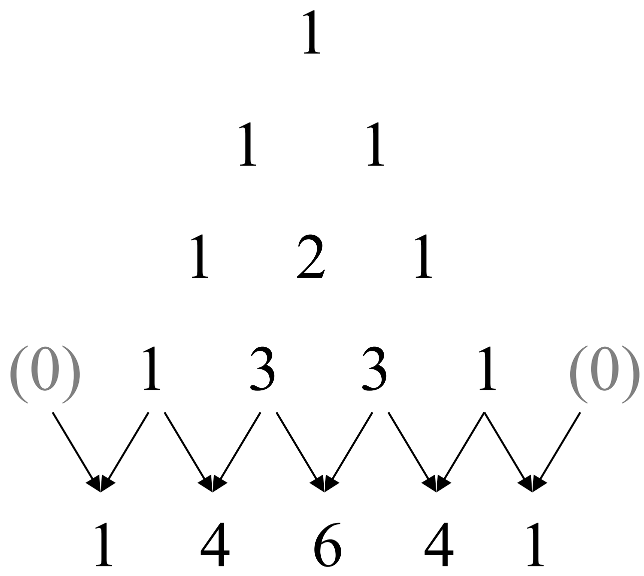
Temporal complexity of the function Pascal

- Let's define the function {Pascal N}

- This function takes a natural number *n* and returns the nth row of Pascal's triangle, represented by a list of integers



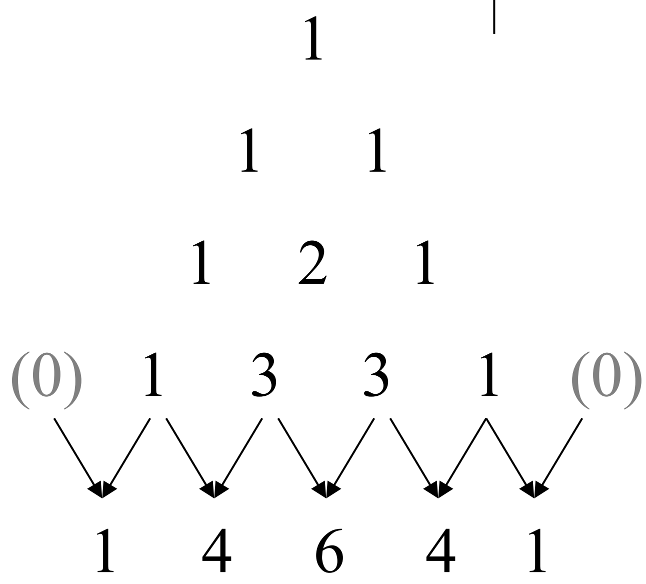
- One way to define the nth row of Pascal's triangle is as the list of coefficients in the expansion of $(a+b)^n$

$$(a+b)^3 = 1a^3 + 3a^2b + 3ab^2 + 1b^3$$

Algorithm to compute Pascal

- Algorithm for {Pascal N}

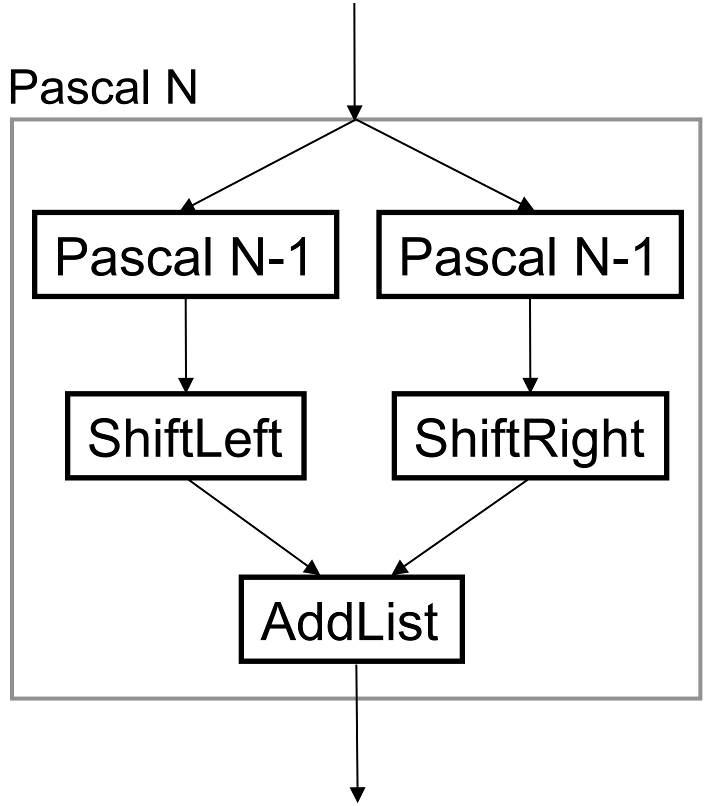
1. For row *0*, return [1]
2. For row *n>0*, shift left row *n-1* and shift right row *n-1*
3. Align the two shifted rows and add them element by element to calculate row *n*



Shift right: [0 1 3 3 1]
Shift left: [1 3 3 1 0]
Add: [1 4 6 4 1]

Source code for {Pascal N}

```
declare
fun {Pascal N}
   if N==0 then [1]
   else
      {AddList
        {ShiftLeft {Pascal N-1}}
        {ShiftRight {Pascal N-1}}}
   end
end
```



Auxiliary functions

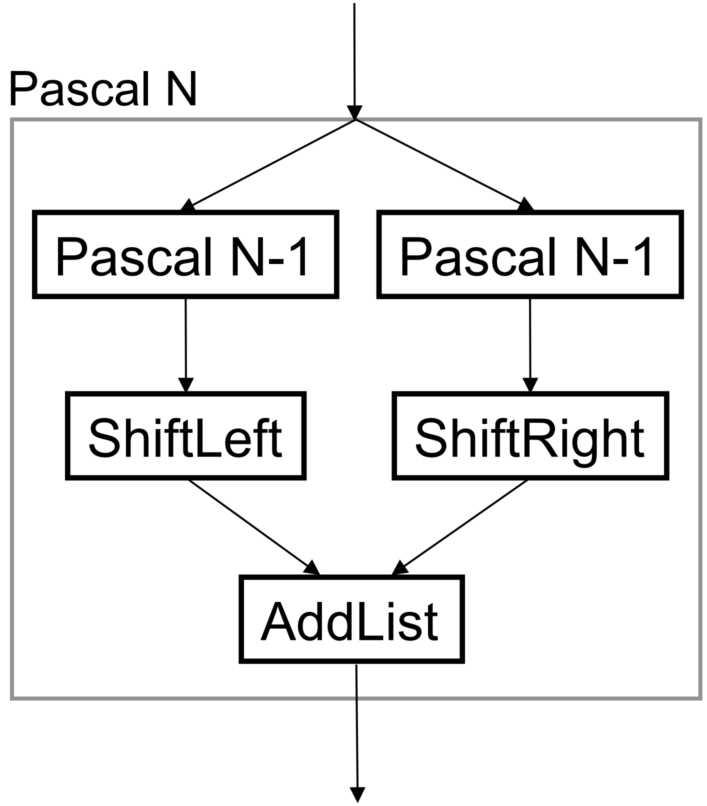
```
fun {ShiftLeft L}
   case L of H|T then
      H|{ShiftLeft T}
   else [0]
   end
end

fun {ShiftRight L}  0|L end
```

```
fun {AddList L1 L2}
   case L1 of H1|T1 then
      case L2 of H2|T2 then
         H1+H2|{AddList T1 T2}
      end
   else nil end
end
```

Temporal complexity of {Pascal N}

```
declare
fun {Pascal N}
   if N==0 then [1]
   else
      {AddList
        {ShiftLeft {Pascal N-1}}
        {ShiftRight {Pascal N-1}}}
   end
end
```



Simplified analysis

- {Pascal N}

  does **2** calls of {Pascal N-1} (if N>0),

  which gives a total of **4** calls of {Pascal N-2},

  …,

  which gives a total of $\mathbf{2^n}$ calls of {Pascal 0}.

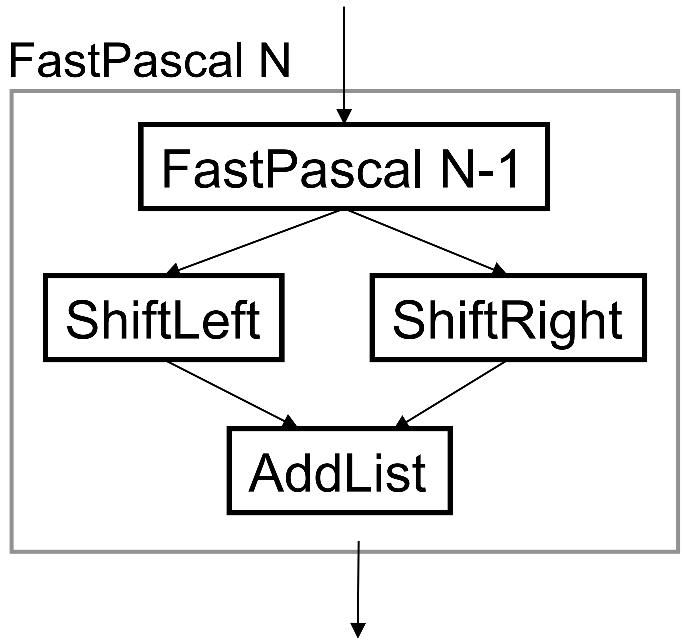
- The temporal complexity is therefore exponential:

  $1+2+2^2+\ldots+2^n \in O(2^n)$
- This is the best (tightest) bound: we can't find any upper bound better (smaller) than 2^n

FastPascal

- We can get by with just one recursive call if we keep the temporary result in a local identifier L

```
fun {FastPascal N}
   if N==0 then [1]
   else L in
      L={FastPascal N-1}
      {AddList {ShiftLeft L} {ShiftRight L}}
   end
end
```



- The complexity is now $n+(n-1)+\ldots+1 \in O(n^2)$
- *Much better!* With FastPascal, we can calculate rows with huge numbers of elements.