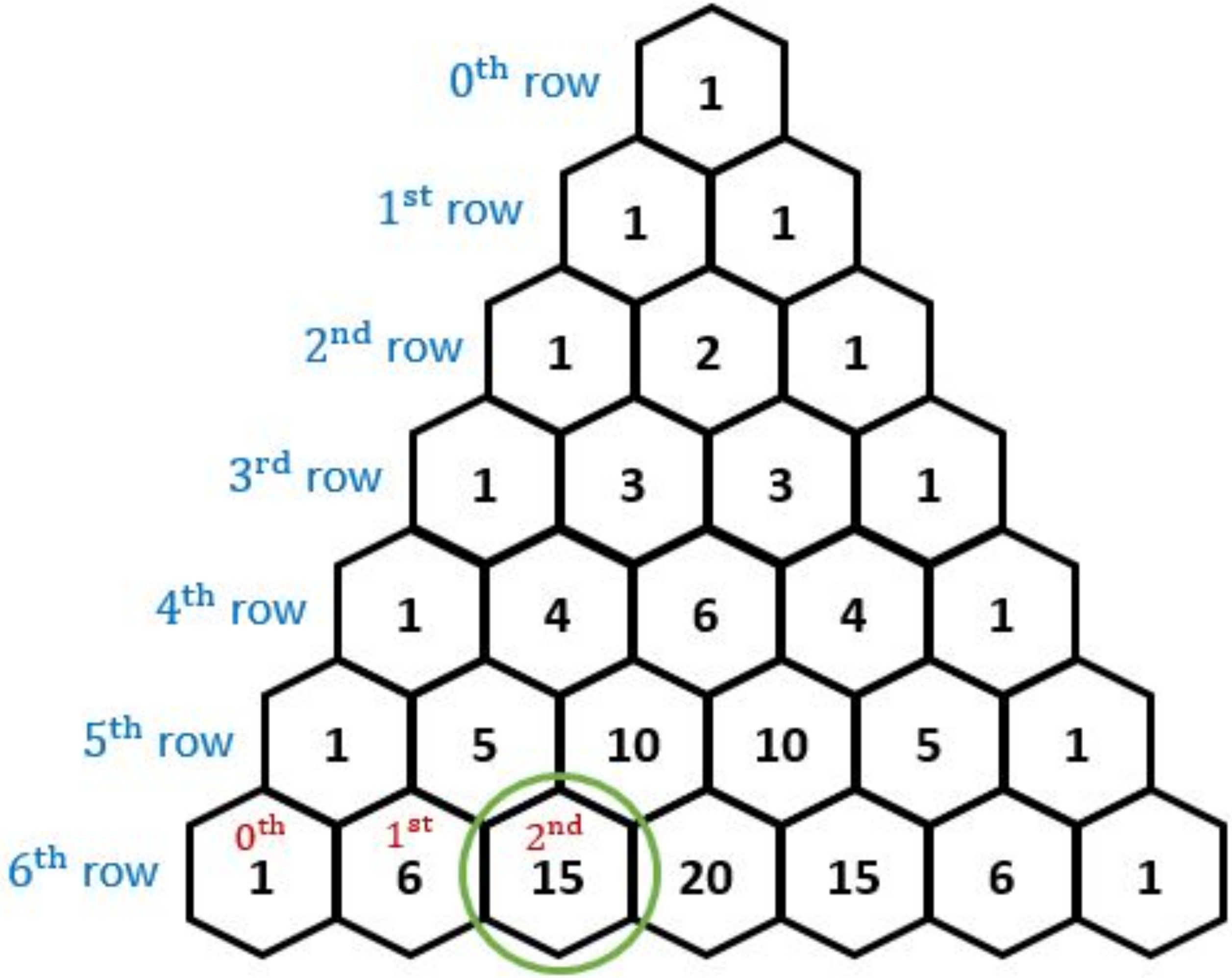
Algorithm Analysis (Example)

CPSC 319 - Data Structures

Benyamin Bashari

Pascal's Triangle



Pascal's Triangle

	0^{th} Column	1^{th} Column	2^{th} Column					
0^{th} Row	1							
1^{th} Row	1	1						
2^{th} Row	1	2	1					
	1	3	3	1				
	1	4	6	4	1			
	1	5	10	10	5	1		
	1	6	15	20	15	6	1	
	1	7	21	35	35	21	7	1

Pascal's Triangle Recursion

- Assume we want to compute the element in n^{th} row and k^{th} column.
- In row i (0 <= i <= n),
  - There are i+1 columns (0 … i)
  - 0-th column is 1
  - i-th column is 1
  - Otherwise j-th column (0 < j < i) is equal to the sum of the upper element and upper left element.

Solving Pascal's Triangle
First Approach: Recursive Functions

- Write a recursive function that computes the element in n-th row and k-th column of pascal's triangle.
- f(n, k) is the number of n-th row and k-th column of pascal's triangle (0 <= k <= n).

$$f(n,k) = \begin{cases} 1 & k = 0 \ or \ n \\ f(n-1,k) + f(n-1,k-1) & otherwise \end{cases}$$

- Compute the runtime of f(n, k) for various n and k, does memoization helps for this function?

Solving Pascal's Triangle Second Approach: Loops

- Instead of using a recursive function, use loops and 2d arrays to compute the f(n, k).
- Compute the runtime of f(n, k) for various n and k, compare it with the first approach

$$f(n,k) = \begin{cases} 1 & k = 0 \ or \ n \\ f(n-1,k) + f(n-1,k-1) & otherwise \end{cases}$$

Solving Pascal's Triangle
Third Approach: Combinatorics

$$f(n,k) = \begin{cases} 1 & k = 0 \; or \; n \\ f(n-1,k) + f(n-1,k-1) & otherwise \end{cases}$$

$$f(n,k) = \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

- Use factorial to compute this expressions, compute the runtime for various n and k. Compare the 3 approaches. Which one is faster?