**Binomial theorem**

Binomial coefficients come from the participation of these numbers in the binomial formula.

Binomial coefficient C (n, k) gives the number of ways in which k objects can be chosen from n objects irrespective of the order.

Formula

C (n, 0) = C (n, n) = 1, for n>=0

C (n, k) = C (n-1, k-1) + C (n-1, k), for n>k>0

For large values of n, same coefficients are computed multiple times. Therefore, if we do not cache values as we compute them, this can take a lot of time because of duplicate computation. This can be avoided using dynamic programming.

Construct a temporary 2D matrix of size n\*k with 1st column and diagonal elements = 1. Fill the table row by row using recursive formula. The table needs to be filled only for n< k. The triangular array of binomial coefficients leads to Pascal’s triangle.

Processing time = O(n\*k)

Memory = O(n\*k)

If we just need to calculate the given combination, then instead of 2D matrix, we can use a 1D array where memory is reduced to O(k+1)