Reflections:

Binomial coefficient

Formula:

C(n,k) = C(n-1,k-1) + C(n-1,k)

1. dynamic programming version-1:
    1. Involves two comparisons.
    2. Creates 2-D array representation.
    3. Doesn't computes the result which are already computed.
    4. Calculates value using previously stored values.
    5. The second for loop runs from j=0 to min (i,k) this is because in case where k reaches k+1 iteration in n side, it should return the value k only.
    6. Doesn't compute the entire 2-D.
    7. Time complexity: O(n*k)
    8. Space complexity: O(n*k)

2. Recursive:
    1. Each recursive call is going to create spaces for variables created during computation.
    2. Computes the same subproblem again and again i.e overlapping subproblem.
    3. Space used: O(2^n – 1) * 8 (bytes)
    4. Time complexity O(2^n-1).

3. Dynamic programming version-2 (using 1-D array):
    1. Display only the final array elements as intermediate results are used to generate final result.
    2. Computes next row of pascal triangle using the previous row.
    3. Every loop on i builds i'th row of pascal triangle using(i-1)th row therefore, 2-D representation is reduced to 1-D.
    4. Time complexity: O(n*k)
    5. Space complexity: O(k)