Dynamic Programming

1. Write a function that uses top-down dynamic programming to calculate the binomial coefficients, based on the recursive rule:

\[
\binom{N}{k} = \binom{N-1}{k} + \binom{N-1}{k-1}
\]

The boundary conditions are:

\[
\binom{N}{0} = \binom{N}{N} = 1
\]

Compute the binomial coefficient for \( N = 30 \) and \( k = 15 \).

What are the performance advantages of an implementation based on top-down dynamic programming compared to an implementation based on bottom-up dynamic programming, for calculating binomial coefficients?

2. The expected number of comparisons performed by the QuickSort algorithm (which we will describe later in the course) when sorting \( N \) elements is:

\[
C(N) = \begin{cases} 
N + \frac{1}{N} \sum_{k=1}^{N} \binom{N-1}{k} + \binom{N-1}{N-k} & N \geq 2 \\
0 & N \leq 1 
\end{cases}
\]

Write a function that uses bottom-up dynamic programming to compute \( C(N) \).

Tabulate \( C(N) \) for \( N = 10, 20, 100, 200 \) and \( 1000 \).