5. Evaluate \[ \frac{1}{N} \sum_{1 \leq k \leq N} \sum_{t} \frac{t(N-k)(k-1)}{(N-1)} \]

6. The **Merge Sort** program sorts \( n \) numbers \( X(1), X(2), \ldots, X(n) \) by:

1. If \( n = 1 \), then do nothing. Otherwise, do Steps 2 through 4.
2. Sort the \( \lfloor n/2 \rfloor \) numbers \( X(1), X(2), \ldots, X(\lfloor n/2 \rfloor) \) by calling **Merge Sort** recursively.
3. Sort the \( \lceil n/2 \rceil \) numbers \( X(\lfloor n/2 \rfloor + 1), X(\lfloor n/2 \rfloor + 2), \ldots, X(n) \) by calling **Merge Sort** recursively.
4. Merge the two sorted lists from Steps 2 and 3. (This takes \( n - 1 \) comparisons in the worst case, using the obvious algorithm.)

Let \( S(n) \) be the worst-case number of comparisons needed by **Merge Sort** to sort \( n \) numbers. The above description of the program shows that

\[
S(n) = S(\lfloor n/2 \rfloor) + S(\lceil n/2 \rceil) + n - 1, \quad n > 1; \\
S(1) = 0.
\]

Solve the recurrence for the general case, for \( n = 1, 2, 3, \ldots \)