Practical Analysis

Assume array A is indexed from 1 to n.

```
INEFFICIENT_SORT($A$, $n$)
$n!$  for $i = 1$ to $n!$ do
          Boolean $sortedSoFar$ = TRUE;
          $j = 1$;
          $P$ = nextPermutation($A$);                  $\Theta(n)$
          while $j < n$ and $sortedSoFar$ do
              if $P[j] > P[j + 1]$                     $O(n)$
              then $sortedSoFar$ = FALSE
              $j$++
          if ($sortedSoFar$) then output $P$           $\Theta(n)$
```

Analyze the worst-case complexity of INEFFICIENT_SORT assuming that the nextPermutation function always takes $\Theta(n)$ time.

$\Theta(n! \cdot n)$

Your answer should fit above the line!