Practical Analysis

Assume array A is indexed from 1 to n.

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

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

Your answer should fit above the line!