

Delay between permutations in ALGORITHM L

Between visiting each permutation, we run at least one **while** loop. These loops take $O(n)$ time in the worst case. Thus, there is an $O(n)$ time delay between visiting permutations

Runtime of ALGORITHM L

For n items, there are $O(n!)$ possible permutations to visit. Then we have $O(n)$ delay between visiting each permutation. Thus, the overall runtime is $O(n \cdot n!)$

Number of decrements of j in ALGORITHM L

Suppose all elements in S are distinct. Exactly half of the possible permutations of S have that $s_{n-1} < s_n$. Thus, the loop will not run for $\frac{n!}{2}$ permutations of S . **Note:** Exercise 1 of [Knu11, Chapter 7.2.1.2] talks about how to modify ALGORITHM L to take advantage of this fact and obtain a slightly faster algorithm.

References

[Knu11] Donald E. Knuth. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Addison-Wesley Professional, 2011. ISBN: 0201038048.