<u>HW.</u> Figure out what 'yield' does in Python.

Thm. Permutations(a, len(a), 0) Prints all Permutations of a.

Pf. By induction on len(a)

* Not a good theorem statement for this implementation.

Thm. Permutations(a, n, start) prints all permutations of a where a[0...(start-1)] is fixed and a[start...(n-1)] is permuted in all possible ways.

Pf. By induction on n-start.

<u>Base case</u> $n - start = 1$. i.e., $start = n-1$. Thm Stmt is $a[0 \ldots n-2]$ is fixed and Print all Permutations of $a[n-1 \ldots (n-1)]$. Observe that the only Permutation that satisfies this is the array a.

Inductive case

$$\underset{\substack{\| \\ k}}{\text{start}} < n-1$$

Inductive hypothesis is the statement for start $= k+1$

The set of all perms is a disjoint union of perms starting with a[start], a[start+1] ..., a[n-1]

the body of the loop brings each of them to the front of the array and uses IH to generate all perms of a[start+1 ... n-1].

$$(1,2,3)$$

$$\{ (1,2,3), (1,3,2) \} \cup \{ (2,1,3), (2,3,1) \} \cup \{ (3,1,2) (3,2,1) \}$$

What could go wrong?.

$a = (0 | 1, 2, 3)$ $n = 4$ Start $= 1$

Print $(0, 1, \boxed{2, 3})$ — $(0, 3, 1, 2)$ X

Print $(0, 1, \boxed{3, 2})$ or $(0, 2, 1, 3)$

$a = (0, 1, 2, 3)$

Thm (final) Permutations(a,n,start)

Prints all perms of a where a[0...(start-1)] is fixed and a[start..(n-1)] is permuted. Moreover, at the end of the function array a is the same as at the beginning.

Pf. By induction on (n-start).

<u>Hw.</u> complete the Proof. I'll upload the whole Proof by next week.

Derangements

A Permutation of $0 \dots (n-1)$ where i is not in Position i for all i.

Ex. $n=3$

$$\begin{array}{ccc} \underline{0} & \underline{1} & \underline{2} \\ 2, & 0, & 1 \\ 1, & 2, & 0 \end{array} \Bigg\} \text{ Only 2 derangements for } n=3$$

Combinations

(a, n, k)

↓

generate all k-combinations of a.