Derangement Problem

Yize Wang

August 3, 2024

1 Problem Statement

Assume there are N birds and N nests both indexed from 1 to N. How many unique permutations are there if each bird wants to live in a nest with a different index from its own. In other words, the k^{th} bird does not want to live in the k^{th} nest.

2 Solutions

An algebraic solution and a brute-force backtrace solution are offered in this section.

2.1 Algebraic Solution

Let D_N denote the number of permutations when there are N birds and N nests.

$N = 1$, the bird has to stay in its own nest, and no permutation qualifies, i.e., $D_1 = 0$;
$N = 2$, the only permutation is that the two birds uses the other nest, i.e., $D_2 = 1$;
$N > 2$, we consider two steps: Let Bird 1 decide first and it has $N-1$ options. Without loss of generality, assume Bird 1 picks Nest 2; Next step, Bird 2 can either pick Nest 1 (case A) or Nest $\{3, \ldots, N\}$ (case B). In case A, Bird $\{3, \ldots, N\}$ should pick their nests from Nest $\{3, \ldots, N\}$, which results in D_{N-2} permutations (sub-problem of $N-2$ birds and nests). In case B, Bird 2 cannot pick Nest 1 which would belong to case A. Then Bird 2 has to pick a nest from Nest $\{3, \ldots, N\}$, and Bird $\{3, \ldots, N\}$ has to choose from $\{1, 3, \ldots, N\}$. If we rename Nest 1 to Nest 2, nothing will be changed for this sub-problem, which is equivalent to solving D_{N-1}.

The permutations in step 1 and step 2 should be multiplied, and thus we obtain the recursive formula:

$$\begin{aligned} D_1 &= 0 \\ D_2 &= 1 \\ D_N &= (N-1)\left(D_{N-1} + D_{N-2}\right), \forall N > 2, N \in \mathbb{Z} \end{aligned}$$

We list below some results for small Ns.

N	1	2	3	4	5	6	7	8	9	10	11	12
D_N	0	1	2	9	44	265	1854	14833	133496	1334961	14684570	176214841

Math masters also derived general formula for D_N. Interested readers are referred to Wikipedia by keyword *Derangement* for more details. Programmers may also want to check-out *Leetcode 634 Find the Derangement of an Array* for more insights.

2.2 Backtrace Solution

We can also utilize modern computer's super power to iterate all possible permutations. The implementation can be found in *main.cpp*. The algorithm's space and time complexity are $\mathcal{O}\left(n^2\right)$ and $\mathcal{O}\left(n \times n!\right)$ respectively.

3 Acknowledgment

Thank Chengzhi Qi brought up this problem and offered insights.