Code Jam 2021 - Qualification Round

Analysis: Reversort Engineering

Test Set 1

The solution to the problem is a permutation of the numbers from 1 to $\mathbf{N}$. The cost of each permutation can be calculated by simulating the Reversort algorithm as described in the analysis of the Reversort problem in $O(\mathbf{N}^2)$ time complexity. There are $\mathbf{N}!$ distinct permutations of size $\mathbf{N}$, containing the numbers from 1 to $\mathbf{N}$ exactly once each. The cost of each permutation can be calculated and the answer is any permutation that has a cost equal to $\mathbf{C}$. If there is no such permutation, output `IMPOSSIBLE`. The time complexity of the overall solution is $O(\mathbf{N}! \cdot \mathbf{N}^2)$.

Test set 2

As $\mathbf{N}$ is large for Test Set 2, we cannot generate every possible permutation. The major observation here is that the range of valid costs for a given $\mathbf{N}$ lies between $\mathbf{N}-1$ (when the cost of each reverse operation is the minimum possible, which is 1) and $\frac{\mathbf{N}\cdot(\mathbf{N}+1)}{2}-1$ (when the cost of each reverse operation is the maximum possible, which is $\mathbf{N}-i$. Cost = $\mathbf{N}-1$ when the array is already sorted.

All costs in between those two limits are possible, as we shall see. Hence, if $\mathbf{C}$ is not in the valid range for given $\mathbf{N}$, output `IMPOSSIBLE`. Otherwise, we perform the following construction by recursion, which also serves as proof that the costs in range are indeed possible. The first iteration costs between 1 and $\mathbf{N}$, so we should choose a cost x for it such that $\mathbf{C}-x$, fits in the possible range for a permutation of size $\mathbf{N}-1$. You can check that this is always possible, and even compute the full range of x values that work by solving the system of inequalities.

Now, recursively generate a permutation P of size $\mathbf{N}-1$ and cost $\mathbf{C}-x$. Then, add 1 to all integers in P and insert 1 at its left end, getting a new permutation of integers between 1 and $\mathbf{N}$. Then, reverse the prefix of P of length x as the cost of the initial iteration should be x. The non-recursive steps take $O(\mathbf{N})$ to adjust P. Since we perform those for each index, the overall complexity of the solution is $O(\mathbf{N}^2)$.