CP Problems Analysis

xyzyzl

4/6/2020

1 Codeforces

1327D. Infinite Path
Think of each permutation mapping as a graph. Let $a \to b$ be an edge if $P_a = b$. It's clear that the minimum value of k s.t $C_{P_x^k} = C_x$ for some x would be the smallest value where the node kx away from the current node has the same color as the current node for all nonnegative integers x. As this is the minimum such k, it follows that $k|n$. Thus, we can loop over all divisors of n and check if it satisfies the condition.
The mistake I made was that I did not remember to check visited for the *current* node, but rather checked it on the *next* node. Thus, the next iteration, it would see that the node was "visited" when it really was not, and break. Thus, all of my answers were just 1.

2 USACO

2020 US Open - Gold: Haircut
This was actually extremely simple and I don't know why I didn't get it.
When I found inversions with the algorithm, for each index i, it counted the number of elements $j < i$ that had $a_j < a_i$ using a BIT. Then, it subtracted that value from i.
So we can find the number inversions for all indices i of a specific value. And that's all we really want. From there we can find the number of inversions solely comprising values $\leq x \forall x$.
Update: I read my code for *mincross* which had inversion count and I feel terrible for not noticing this – it was literally in the code.

2020 US Open - Gold: Exercise
So I made significant progress during the contest but for some reason just didn't implement it. A few key ideas that I came up with were that:

- Each permutation is a "cycle" of different permutations. The number of times before repetition is the lcm of all cycle lengths.

- We need to find $lcm(a_1, a_2, \ldots, a_g)$ for some g s.t. $\sum_{i=1}^{g} a_i = n$.

- Anything that works for $j < i$ steps also works for i steps.

However, I did not know how to find the lcm of all a_i despite it being clear from the third observation that it was DP.
Let $dp_{i,j}$ be the running sum with the first i primes and the first j indices. Then

$$dp_{i,j} = dpi - 1, j + (p_i^a \cdot dp_{i-1,j-p_i^a} \forall a \leq \log_{p_i} j).$$

$dp_{i-1,j}$ is the previous step and the other long expression is the added running sum from the ith prime p_i.