Problem Tutorial: "Joke"

Firstly, $f(p, q)$ clearly depends only on the order of pairs (p_i, q_i), so we can assume that $p_i = i$ initially.

Lemma. $f((1, 2, \ldots, n), q)$ = number of increasing subsequences of q.

Proof. Let's first analyze when the string s satisfies p, q. Basically, we have n nodes corresponding to upper row, n nodes corresponding to lower row, and some directed edges (u, v) between them, indicating that the number in cell u has to be smaller than the number in cell v. The necessary and sufficient condition for being able to put numbers from 1 to $2n$ into this nodes so that all relations are satisfied is: **There has to be no directed cycle.** We will show that in case of our graph, it's equivalent to the following: **There exists no directed cycle of size** 4.

Indeed, consider the directed cycle of the smallest length, suppose that its size is larger than 4. It has to contain some edge between nodes from two different rows, as there can't be any cycle inside a single row. Wlog it's an edge from cell $(1, i)$ to $(2, i)$. There has to be an edge from $(2, i)$ somewhere now, wlog to $(2, j)$. Finally, if the edge from $(2, j)$ goes to $(2, k)$, we could have obtained a shorter cycle by just removing $(2, j)$ from it, as there is an edge $((2, i), (2, k))$, so the edge from it goes to $(1, j)$. Now, if $p_i < p_j$, then we can replace the path $((1, i), (2, i), (2, j), (1, j))$ by just $((1, i), (1, j))$, otherwise we have obtained a cycle of size 4.

So, it's enough to ensure that there are no directed cycles of size 4. Let's find the number of strings s for which it's the case. Consider i for which $q_i = n$. If we set s_i to `0`, we can forget about pair (p_i, q_i), as it can't be involved in any cycle of length 4. Otherwise, we get that the number in the cell $(1, i)$ of the matrix is bigger than the largest number in the second row, so for each $j > i$, the number in cell $(1, j)$ is also bigger than in cell $(2, j)$. Therefore, if we set s_i to `1`, we also have to set all s_j with $j > i$ to `1`. After that, we can throw out all pairs (p_j, q_j) for $j \geq i$, as there wouldn't be able to get involved in any cycles.

So, we have an array q, and 2 operations:

- Delete the largest element
- Delete the largest element and all elements to the right of it.

It's easy to show that the number of ways to delete the entire q by applying these operations in some order is equal to the number of increasing subsequences of q. Indeed, each such sequence of operations corresponds to the subsequence of numbers to which we will apply 2-nd operation, when they are the largest.

Lemma is proved

Now, we have the following problem:

- We are given some elements of permutation q, and others are missing. Find sum of $f(q)$ over all valid permutations q (meaning that they have the given elements at the right places).

Under $n \leq 100$, it's an easy problem. Set $q_0 = 0$ and $q_{n+1} = n + 1$, now $f(q)$ is the number of increasing subsequences starting at q_0 and ending at q_{n+1}. For every element that's already set, say q_i, calculate $dp[i][k]$ — the number of possible increasing subsequences starting at q_0 and ending at q_i, which contain exactly k **unset** elements.

Here are the transitions: for every $j < i$ such that q_j is also set and $q_j < q_i$, we calculate the number of "free" positions between jth and ith, and the number of "allowed" elements — the elements from $[q_j + 1, q_i - 1]$, which aren't set as elements already. Then, for every *choose* not exceeding $max(free, allowed)$ and every *chosen*, add $dp[j][chosen] \times (\binom{choose}{allowed} \times \binom{choose}{free})$ to $dp[i][chosen + choose]$.

The answer to the problem is then just the sum of $dp[n + 1][x] \times (n - set - x)!$ over x, where set is the number of already set elements.