Problem F

Spent a few days on and off having this problem at the back of my mind.

I think this problem was difficult but it wasn't unapproachable. When I first read the problem, I wondered if there was even an impossible case.

The idea of the problem is there, if you list out all the inversions. If you swap all of them pairwise then there is always a sequence which would resolve all the inversions. As long as when you swap you don't create any more inversions, using all of them and resolving one by one, you end up removing all the inversions present.

I then noticed the input, the range of input goes only up to $n \leq 2000$ and the idea of the problem is your are swapping elements pairwise. This tends to either a $O(n^2)$ or $(n^2 \log n)$ solution. A well known algorithm which swaps elements while sorting in $O(n^2)$ is bubble sort and so I considered that as a possibility. An algorithm which mimics bubble sort in order to solve the problem.

After that, I didn't really think about it much but decided to come back to it at 2am one night. I noticed that the actual value of the numbers don't matter but just the relative ordering. As such you can compress the input to $1 \ldots n$ this makes it easy to sort/compute.

You would then just sort this new array of $1 \ldots n$ in a way which mimics the ordering given by the original array.

Basically what happens here is you bubble sort n up to the top (last element) and record any inversions you "fix" as you go. You would then repeat this algorithm for $n-1, n-2, \ldots$

Turns out this fixes all inversions in a correct order.

Another method you could do is to just find all the inversions, then just sort the pair of inversions in order of decreasing right element, and on ties with increasing left elements. For a pair of indicies (i, j) for $i < j$.