Fancy Stack

Problem author: Dmitry Gozman; problem developer: Gennady Korotkevich

Let's call the blocks on the even positions *big* $(b_2, b_4, \ldots, b_n)$, and the blocks on the odd positions *small* $(b_1, b_3, \ldots, b_{n-1})$.

First, assume that the block sizes are distinct. We'll process the blocks in decreasing order of size (i.e., let $a_1 \geq a_2 \geq \ldots \geq a_n$). To count the stacks, we will use dynamic programming.

Let $f_{i,j}$ be the number of ways to put i biggest blocks into their places so that j of these blocks are big, and $i - j$ of these blocks are small. As a base case, $f_{0,0} = 1$. We'll implement it as a forward DP, and to make transitions for the $i + 1$-th block we will decide whether it's big or small.

If the $i + 1$-th block is big, its position in the stack is determined uniquely (specifically, it's b_{n-2j}). Hence, we make a transition to $f_{i+1,j+1}$.

If the $i + 1$-th block is small, there are $\max(j - 1, 0) + [j = \frac{n}{2}]$ possible places for small blocks (between any two big blocks, and at the top of the stack if all big blocks have been placed), out of which $i - j$ are already occupied. Note that all these potential places will be available for all future (smaller) small blocks. Hence, it doesn't matter which particular place we occupy with the $i + 1$-th block. We make a transition to $f_{i+1,j}$ with coefficient $(\max(j - 1, 0) + [j = \frac{n}{2}]) - (i - j)$.

Now, if we allow blocks to have equal sizes, we can just process the blocks in groups of the same size — again, in decreasing order of size. In each group, at most one block can be large (since all big blocks must have distinct sizes). This way, we still have just two transitions from any DP state, this time with both of them using binomial coefficients — coming from the fact that we can choose any valid unoccupied places for small blocks.

Alternatively, instead of splitting the blocks into groups explicitly, we can keep the solution from the "all distinct" case, and only allow the last block in each group to be big (that is, block i can only be big if $a_i > a_{i+1}$).

The time complexity of this solution is $O(n^2)$.