Analysis of task tree2

Solutions, statement, analysis: Dimitar Shapatov; Solutions, analysis: Kiril Zashev

Subtask 1

The biggest hint here for the contestants should be the constraints. They are the same as the ones from the task Magic Trick from the IOI 2025 Practice Session. For those who do not know this task – ☹. In that task one has to place a maximum number of -1s on a permutation so that it can afterwards be decoded successfully (it reminds me of another task, but that is a separate question...) In order to solve the task, one has to apply the Erdős-Szekeres theorem. It states that every sequence of $N \geq (a-1)(b-1)+1$ distinct real numbers certainly contains an increasing subsequence of length a, or a decreasing subsequence of length b. This is the key observation in the task, but a problem arises – in `decode` we do not know whether we have taken an increasing or a decreasing subsequence. We can solve this problem easily. We know that if the longest increasing one is of size below $\lfloor\sqrt{N}\rfloor$, then the longest decreasing one is of size at least $\lfloor\sqrt{N}\rfloor+1$, therefore if the longest increasing subsequence is of size at least $\lfloor\sqrt{N}\rfloor$, we take its first $\lfloor\sqrt{N}\rfloor$ elements, and otherwise we take $\lfloor\sqrt{N}\rfloor+1$ elements of the longest decreasing subsequence. This way we distinguish which of the two we have taken, depending on whether the number of -1s is $\lfloor\sqrt{N}\rfloor$ or $\lfloor\sqrt{N}\rfloor+1$.

But how does this help us in this task? It turns out that we can reduce our task to a task about a permutation if we do an Euler tour of the tree, keeping the vertices only when they are entered. In the decoding there will be no problem returning the same Euler tour as in the encoding, because the edges are given in the same order in which they were given to `encode` – this way the adjacency lists are built in the same order and the traversal is identical. The different removed vertices are given as different negative numbers, so this is also not a problem – we can distinguish them. We also have no problems with the root, since it is always at 1 or -1 and we can simply use whichever of the two we have.

Subtask 2

The bigger constraint simply bothers us with having to do the fast way of finding the longest increasing subsequence and the longest decreasing sequence through a segment tree. Here the implementation is even more annoying.