Sequence

Subtask1

- We can use a simple brute force algorithm.
- For example, enumerate over every possible (l, r) and sort(or nth_element) the sequence to find the median. Then count the occurrence.
- Time complexity: $O\left(n^3\right) - O(n^3 \log n)$.

Subtask2

- Here we need to find medians of a sequence quickly.
- A simple way is to use two heaps to maintain the bigger part and the smaller part of the sequence. Or just use a balanced tree.
- Time complexity $O(n^2 \log n)$.
- There's also another way, reverse the operations, then we just need to delete an existing value and find the medians. This can be done with a linked list in $O(n^2)$.

Subtask3

- For every value x, occurrences of x form two continuous subsequence.
- If we choose x as the final median. We either choose one of the subsequence or choose both if possible.
- It’s easy to determine whether it’s possible to choose both.

Subtask4

- If the median of the whole sequence is not 2, then it appears in the sequence more than $n/2$, and that's of course the answer. For the rest of the cases, choosing $l = 0, r = n - 1, x = 2$ is the best choice if $x = 2$.
- Then just consider $x = 1,3$. If we choose $x = 1$, we can see every 1 as 1, and every 2,3 as -1. Then we want to find a continuous subsequence with non-negative sum and with the most 1.
- Do a prefix sum, then for every r find the smallest l satisfying $sum_r \geq sum_l$.

Subtask5

- We just need to determine whether the answer is 1 or 2.
- For every value x that appeared twice at x_1, x_2:
- Construct a sequence $B_i = \max(-1, \min(A_i - x, 1))$.
- If we can find $l \leq x_1 \leq x_2 \leq r$ with sum $[-2,2]$, then x is a median of such (l, r).
- Find $l \leq x_1 \leq x_2 \leq r$ with the biggest sum and smallest sum, call these sums v_1, v_2. Since moving l or r by one step only affect the sum by at most 1, so every integer sum in $[v_2, v_1]$ is achievable.

Subtask 6&7

- From the solution of Subtask5, we can actually determine if there is (l, r) satisfying $l \le x_1 \le x_2 \le r$ and has median x.
- View every $z \ge x$ as 1, others as -1, find (l, r) with maximum sum, call this v_1.
- View every $z > x$ as 1, others as -1, find (l, r) with minimum sum, call this v_2.
- If $v_1 * v_2 \le 0$, then there exist such (l, r).

Subtask 6&7

- Enumerate x from small to big.
- For the occurrences of x. Run two pointers to see if there is a sequence with median x and contains every x from l to r. The checking is described before.

- Time complexity: $O(n \log n)$.