961. N-Repeated Element in Size 2N Array

Easy | Topics | Companies

You are given an integer array `nums` with the following properties:

- `nums.length == 2 * n`.
- `nums` contains `n + 1` **unique** elements.
- Exactly one element of `nums` is repeated `n` times.

Return *the element that is repeated* `n` *times.*

Example 1:

```
Input: nums = [1,2,3,3]
Output: 3
```

Example 2:

```
Input: nums = [2,1,2,5,3,2]
Output: 2
```

Example 3:

```
Input: nums = [5,1,5,2,5,3,5,4]
Output: 5
```

Constraints:

- `2 <= n <= 5000`
- `nums.length == 2 * n`
- `0 <= nums[i] <= 10^4`
- `nums` contains `n + 1` **unique** elements and one of them is repeated exactly `n` times.

Python:

```
class Solution:
    def repeatedNTimes(self, A: list[int]) -> int:
        for i in range(len(A) - 1):
            if A[i] == A[i + 1] or (i + 2 < len(A) and A[i] == A[i + 2]):
                return A[i]
        return A[0]
```

JavaScript:

```
const repeatedNTimes = A => {
    for (let i = 0; i < A.length - 1; ++i) {
        if (A[i] === A[i + 1]) return A[i]
        if (A[i] === A[i + 2]) return A[i]
    }

    return A[0]
};
```

Java:

```
class Solution {
    public int repeatedNTimes(int[] A) {
        for (int i = 0; i < A.length - 1; i++)
            if (A[i] == A[i + 1] || (i + 2 < A.length && A[i] == A[i + 2]))
                return A[i];

        return A[0];
    }}
```