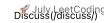
<sup>Day 3</sup> Problems(/problemset/all/)

Interview Contest/)











# 5800. Build Array from Permutation

My Submissions (/contest/weekly-contest-248/problems/build-array-from-permutation/submissions/)

Back to Contest (/contest/weekly-contest-248/)

Given a zero-based permutation nums (0-indexed), build an array ans of the same length where ans[i] = nums[nums[i]] for each  $0 \le i \le nums.length$  and return it.

A **zero-based permutation** nums is an array of **distinct** integers from  $\emptyset$  to nums.length -1(inclusive).

## User Accepted: 64 **User Tried:** 73 **Total Accepted:** 64 **Total Submissions:** 72 Difficulty: Easy

#### Example 1:

```
Input: nums = [0,2,1,5,3,4]
Output: [0,1,2,4,5,3]
Explanation: The array ans is built as follows:
ans = [nums[nums[0]], nums[nums[1]], nums[nums[2]], nums[nums[3]], nums[nums[4]]
    = [nums[0], nums[2], nums[1], nums[5], nums[3], nums[4]]
    = [0,1,2,4,5,3]
```

#### Example 2:

```
Input: nums = [5,0,1,2,3,4]
Output: [4,5,0,1,2,3]
Explanation: The array ans is built as follows:
ans = [nums[nums[0]], nums[nums[1]], nums[nums[2]], nums[nums[3]], nums[nums[4]], nums[nums[5]]]
    = [nums[5], nums[0], nums[1], nums[2], nums[3], nums[4]]
    = [4,5,0,1,2,3]
```

### **Constraints:**

- 1 <= nums.length <= 1000
- 0 <= nums[i] < nums.length
- The elements in nums are distinct.

```
JavaScript
                                                                                                                  \mathfrak{C}
1 v const buildArray = (a) ⇒ {
2
        let n = a.length;
3
        let res = Array(n).fill(-1);
4
        for (let i = 0; i < n; i++) res[i] = a[a[i]];
5
        return res;
6
    };
```

United States (/region)