Next permutation

arr[] = [3 1 2]

123 213 312
132 231 321

Edge case → 123

① Generate all sorted
② Linear Search
③ .

[2, 1, 5, 4, 3, 0, 0]

Longer prefix match →

a[i] < a[i+1]

① find >1, but the smallest
1

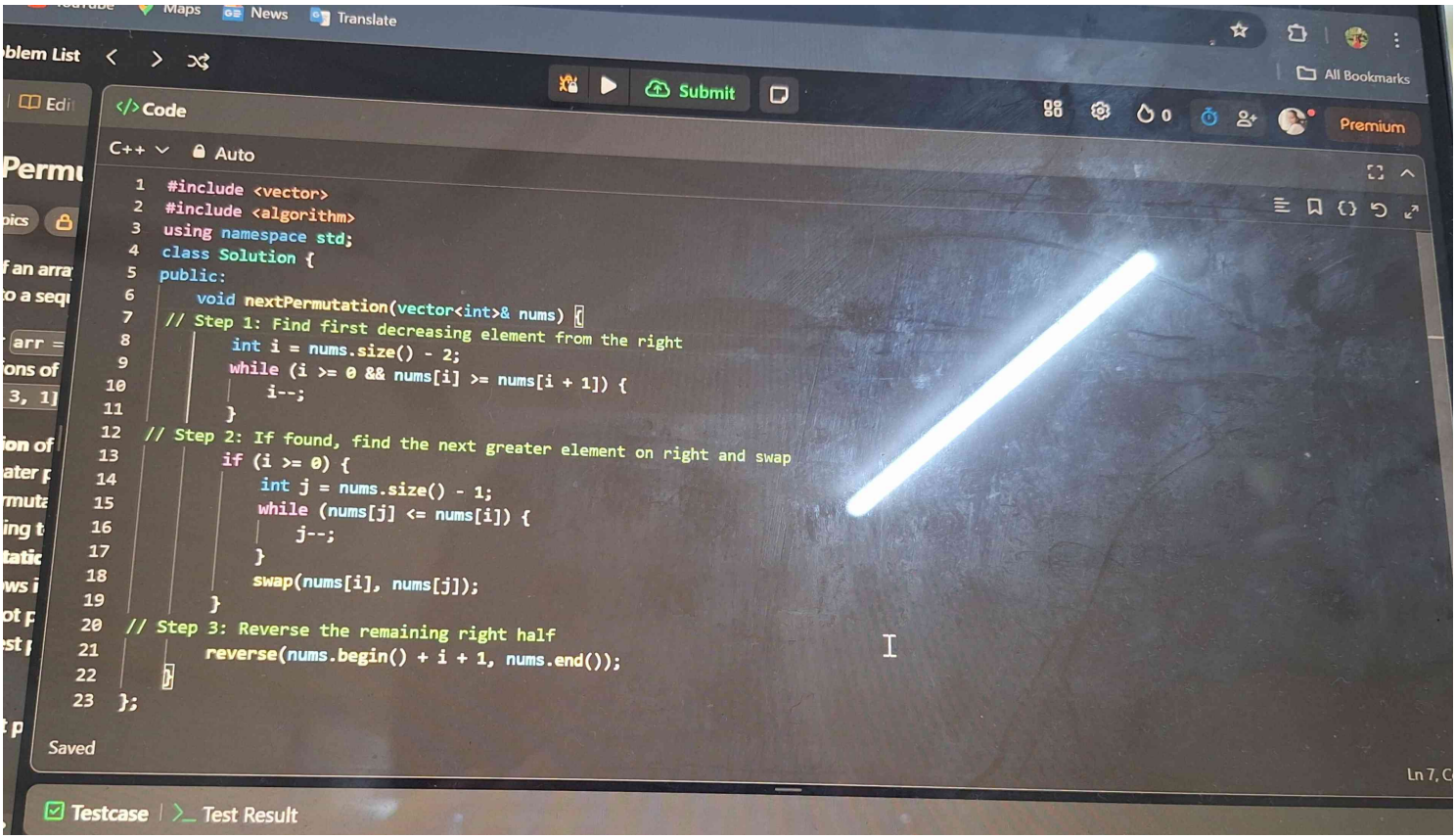
so that you stay close

③ Try to place in sorted order

[1 2 3 4 5]
n-2

[5 4 3 2 1]

```
for (i=n-2 ; i>=0 ; i--)
    if (a[i] < a[i+1]
        { ind = i
          break
        }
}
for (i=n-1 ; i>=ind ; i++)
    { if (arr[i] > arr[ind]
        swap arr[i], arr[ind]
        break;
    }
rev (ar, ind+1, n-1)
```


Maps
News
Translate
Submit
All Bookmarks
Premium
Code
C++
Auto
#include <vector>
#include <algorithm>
using namespace std;
class Solution {
public:
void nextPermutation(vector<int>& nums) {
// Step 1: Find first decreasing element from the right
int i = nums.size() - 2;
while (i >= 0 && nums[i] >= nums[i + 1]) {
i--;
}
// Step 2: If found, find the next greater element on right and swap
if (i >= 0) {
int j = nums.size() - 1;
while (nums[j] <= nums[i]) {
j--;
}
swap(nums[i], nums[j]);
}
// Step 3: Reverse the remaining right half
reverse(nums.begin() + i + 1, nums.end());
}
};
Saved
Testcase
Test Result

Next permutation

We are given a seq of nos.
We need to rearrange them into the next lexicographically greater permutation.

Input → [123] output → [132]
[321] → [123].

nums[i] < nums[i+1] —— first index i from right where.

[1, 2, 3, 6, 5, 4] → i = 2 (first dip)
nums[2] = 3 → 3 < 6

[This i – indicates next permutation]

swap nums[i] with the smallest no. greater than it to the right.
j from right where nums[j] > nums[i]
nums[5] = 4
swap → [1, 2, 4, 6, 5, 3].

After swapping reverse it to make it in the smallest possible order (ascending) ensuring it's the immediate next permutation.

[1, 2, 4, 6, 5, 3]
Reverse part after index = i = 2 [6, 5, 3] → [3, 5, 6]
[1, 2, 4, 3, 5, 6]