1. Search in Rotated Sorted Array II

Medium

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.

(i.e., [0,0,1,2,2,5,6] might become [2,5,6,0,0,1,2]).

You are given a target value to search. If found in the array return true, otherwise return false.

**Example 1:**

Input: nums = [2,5,6,0,0,1,2], target = 0  
Output: true

**Example 2:**

Input: nums = [2,5,6,0,0,1,2], target = 3  
Output: false

**Solution**

step1: find the rotation point using linear search

step2: recover the original array

step3: use binary search

class Solution {  
public:  
 bool search(vector<int>& nums, int target) {  
 if(nums.size() == 0)return false;  
 int pos = search\_point(nums);  
 reverse(nums.begin(), nums.begin() + pos + 1);  
 reverse(nums.begin() + pos + 1, nums.end());  
 reverse(nums.begin(), nums.end());  
 return binary\_search(nums.begin(), nums.end(), target);  
 }  
 int search\_point(vector<int>& nums){  
 int pos = 0;  
 while(pos < nums.size() - 1){  
 if(nums[pos] <= nums[pos+1])pos++;  
 else break;  
 }  
 return pos;  
 }  
};