

旋转数组后求最小值

题目： 有序的数组，从某个位置旋转后，给一个target数，要求在log(n) 的时间内，找出该 target 在数组中的位置。

这种方法包含了： 1) 0个元素旋转，即数组本来就有序的情况。

2) 如果数组有重复元素，暂时没有考虑。【有重复的，用顺序查找】

```
#include <bits/stdc++.h>
using namespace std;

void binarySearch(vector<int>& vec, int target,int& pos, int left, int
right){
    if( left > right )
        return;
    int mid = left + (right - left)/2;
    if( vec[mid] == target ){
        pos = mid;
        return;
    }

    if( vec[mid] > target ){
        /// 判断是否在前半部分
        if( vec[left] > target )
            binarySearch( vec, target ,pos, mid+1, right );
        /// 判断是否在后半部分
        else
            binarySearch( vec, target,pos, left, mid -1 );
    }

    /// target 比中间的还大
    else if( vec[mid] < target )
        binarySearch( vec, target ,pos, mid+1, right );

    return;
}

int main(){
    vector<int> vec{ 1,2,3,4};
    int pos;
    binarySearch(vec, 3, pos, 0, vec.size()-1 );
    cout << pos;
```

```
    return 0;  
}
```