

## 1240：查找最接近的元素

### 题目描述

在一个非降序列中，查找与给定值最接近的元素。

### 输入

第一行包含一个整数 $n$ ，为非降序列长度。 $1 \leq n \leq 100000$ 。

第二行包含 $n$ 个整数，为非降序列各元素。所有元素的大小均在 $0 \sim 1,000,000,000$ 之间。

第三行包含一个整数 $m$ ，为要询问的给定值个数。 $1 \leq m \leq 10000$ 。

接下来 $m$ 行，每行一个整数，为要询问最接近元素的给定值。所有给定值的大小均在 $0 \sim 1,000,000,000$ 之间。

### 输出

$m$ 行，每行一个整数，为最接近相应给定值的元素值，保持输入顺序。若有多多个值满足条件，输出最小的一个。

### 输入样例

```
3
2 5 8
2
10
5
```

### 输出样例

```
8
5
```

## 分析

本题是二分搜索的基本应用，即给定一组数字，再给出多个目标数字，寻找与目标数字最接近的值。

在样例中，第一个目标数字是10，但是它远大于右边界，所以返回了右边界的数值8。

第二个目标数字是5，正好存在于给定的数组中，因此正常返回。

通过样例的提示，我们可以想到如下三点：

- 1、如果所要查询的目标数字超出了给定数组的两个边界，那么我们直接返回两边的边界值即可。
- 2、如果目标数字存在于给定的数组中，则直接返回该数值。
- 3、假设我们使用闭合区间的方式进行搜索，那么最终的结果一定是两个不相等的左右边界值，即 $left \neq right$ ，此时，我们只需要计算二者与目标值的绝对值，数值较小者胜出。

综上，代码如下：

在这里，我们使用闭合区间的搜索方式进行书写，同学们可以自行修改成其他方式。

```
#include <bits/stdc++.h>

using namespace std;
int nums[100001];
int n;

//二分查找法
int BinSearch(int x) {
    int left = 0, right = n - 1;
    int middle;
    //先判断目标数字是否存在于左右两个边界
    if (x <= nums[left]) {
        return nums[left];
    }
    if (x >= nums[right]) {
        return nums[right];
    }

    while (left <= right) {
        middle = left + (right - left) / 2;
        //直接找到目标
        if (nums[middle] == x) {
            return x;
        }
        //中间值大于目标，在左侧进行查找
        else if (nums[middle] > x) {
            right = middle - 1;
        }
        //中间值小于目标，在右侧进行查找
        else if (nums[middle] < x) {
            left = middle + 1;
        }
    }
}
```

//找不到的时候作比较一下到底哪个离得更近;

```
if (abs(nums[left] - x) < abs(x - nums[right])) {  
    return nums[left];  
} else {  
    return nums[right];  
}  
}
```

```
int main(int argc, char **argv) {  
    int t, m;  
    cin >> n;  
    //数组从0开始存储  
    for (int i = 0; i < n; ++i) {  
        scanf("%d", &nums[i]);  
    }  
    cin >> m;  
    for (int i = 1; i <= m; ++i) {  
        cin >> t;  
        printf("%d\n", BinSearch(t));  
    }  
    return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

