

二分函数

T E A C H I N G C O U R S E W A R E P O W P O I N T

授课时间：2025.08.04

目录

- PART-01 函数引入 TEACHING COURSEWARE
- PART-02 函数使用 TEACHING COURSEWARE
- PART-03 经典例题 TEACHING COURSEWARE
- PART-04 总结 TEACHING COURSEWARE

01

函数引入

TEACHING
COURSEWARE

TEACH



通用模版

lower_bound 和upper_bound

数组必须是有序的

02

函数使用

TEACHING
COURSEWARE

TEACH



函数使用

两个函数都是在排好序的数组里快速找位置（类似查字典找单词的位置）：

lower_bound：找第一个大于等于目标数的位置

upper_bound：找第一个大于目标数的位置



函数使用

假设我们有一个从小到大排好序的数组，比如 `arr = [0, 2, 4, 6, 8]`（这里下标 1 到 4 是有效数据，0 号位置不用）。

步骤 1：确定查找范围

数组从下标1开始存数据，总共有n个数据，那么范围就是：
从 `arr+1` 开始，到 `arr+n+1` 结束（记住这个固定写法）

步骤 2：调用函数找位置

用函数找到位置后，减去数组名`arr`，得到的就是下标。

```
int id = lower_bound(arr+1, arr+n+1, x) - arr;
```

03

经典例题

TEACHING
COURSEWARE

TEACH

经典例题

题目背景

[复制 Markdown](#) [展开](#) [进入 IDE 模式](#)

出题是一件痛苦的事情！

相同的题目看多了也会有审美疲劳，于是我舍弃了大家所熟悉的 A+B Problem，改用 A-B 了哈哈！

题目描述

给出一串正整数数列以及一个正整数 C ，要求计算出所有满足 $A - B = C$ 的数对的个数（不同位置的数字一样的数对算不同的数对）。

输入格式

输入共两行。

第一行，两个正整数 N, C 。

第二行， N 个正整数，作为要求处理的那串数。

输出格式

一行，表示该串正整数中包含的满足 $A - B = C$ 的数对的个数。

输入输出样例

输入 #1

[复制](#)

```
4 1
1 1 2 3
```

输出 #1

[复制](#)

```
3
```

说明/提示

对于 75% 的数据， $1 \leq N \leq 2000$ 。

对于 100% 的数据， $1 \leq N \leq 2 \times 10^5$ ， $0 \leq a_i < 2^{30}$ ， $1 \leq C < 2^{30}$ 。

2017/4/29 新添数据两组

```
#include <iostream>
#include <algorithm>
using namespace std;
const int N = 2e5 + 10;
int a[N];
int main() {
    int n, c;
    cin >> n >> c;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }
```

```
// 排序数组
sort(a + 1, a + n + 1);
long long ans = 0;
// 步骤3: 遍历每个B元素
for (int i = 1; i <= n; i++) {
    int b = a[i];
    int target = b + c; // 计算需要的A值
    // 用lower_bound找第一个>=target的位置
    // 减去数组起始地址得到整数位置 (从1开始)
    int l = lower_bound(a + 1, a + n + 1, target) - a;
    // 用upper_bound找第一个>target的位置
    int r = upper_bound(a + 1, a + n + 1, target) - a;
    // 两个位置的差值就是符合条件的A的数量
    ans += (r - l);
}
cout << ans << endl;
```

要找满足 $A-B=C$ 的数对，其实就是找每个 B 对应的 $A=B+C$ 的数量。

我们可以先排序数组，再用 `lower_bound` 和 `upper_bound` 快速定位每个目标值的范围，计算数量总和。

04

总结

TEACHING
COURSEWARE

TEACH