

本节内容：二分查找、三分查找、二分答案、交互二分

## 二分查找

二分查找（英译：binary search），也称折半搜索，是用来在一个**有序数组**中查找某一元素的算法。

**时间复杂度**：二分搜索过程中，算法每次都把查询的区间减半，所以对于一个长度为  $n$  的数组，至多会进行  $\log n$  次查找。

**整数二分例题**：[洛谷]P2249 【深基13.例1】查找

模板1:

```
bool check(int mid) {  
    /*.....*/  
}  
int main()  
{  
    int l = 1, r = n;  
    while (l < r)  
    {  
        int mid = l + r >> 1; // 区间[l, r]被划分成[l, mid]和[mid + 1, r]:  
        if (check(mid)) l = mid + 1;    // check()判断mid是否满足性质  
        else r = mid;  
    }  
    // [l, l-1]为check函数返回为true的部分, [l, r]为check函数返回为false的部分  
    cout << l-1 << endl; // 自己需要调边界  
}
```

模板2:

```
int main() {  
    int l = 1, r = n;  
    while (l <= r)  
    {  
        int mid = l + r >> 1;  
        if (check(mid)) l = mid + 1;  
        else r = mid - 1;  
    }  
}
```

## 浮点二分

```
double eps = 1e-5;  
while(fabs(r-l)>eps) //需要一个精度保证  
{  
    double mid = (l+r)/2;  
    if(check(mid)) l=mid;  
    else r=mid;  
}
```

## 三分查找

通常求单峰(谷)函数

**实数三分例题：**[洛谷]P3382 【模板】三分法

**模板1：**

```
//l表示区间左端点，r表示区间右端点
while(fabs(r-l)>eps) {
    double mid1 = ( r - l ) / 3 + l;
    double mid2 = ( r - l ) / 3 * 2 + l;
    if(f(mid1)<f(mid2)) l = mid1;
    else if(f(mid2)<f(mid1)) r = mid2;
    else break;
}
```

**模板2：**

```
//l表示区间左端点，r表示区间右端点
while(fabs(r-l)>eps) {
    double mid = (l + r) / 2;
    if(f(mid-eps)<f(mid+eps)) l = mid;
    else r = mid;
}
```

## 交互二分

**例题：**[洛谷] P1733 猜数

清空缓冲区

```
// #define endl '\n'
// 宏定义要去掉
cout << endl;
fflush(stdout);
```

## 二分答案

常见题目类型，核心是写**check函数**

**例题：**[洛谷] P2678 跳石头

**例题：**[洛谷] P1824 进击的奶牛

解题的时候往往会考虑枚举答案然后检验枚举的值是否正确。若枚举的可能答案满足**单调性**，则满足使用二分的条件。把这里的枚举换成二分，就变成了「二分答案」。

将枚举答案的时间复杂度 $O(n * n)$ 优化为二分答案的 $O(\log n * n)$

**最大值最小化**（非常重要）

最小值最大化是同理的

## Tips

### 二分函数

```
#include<algorithm> // 二分函数头文件
int main() {
    vector<int> a;
    int b[N];
    set<int> s;
    //找升序数组中d的位置
    lower_bound(a.begin(), a.end(), d);
    lower_bound(b+1, b+1+n, d);
    s.lower_bound(d);
    //升序数组中第一个大于等于d的元素的位置，如果没找到就是a.end()
    upper_bound(); // 升序数组中第一个大于d的元素的位置
}
```

上面讲的是最正确的二分写法，一般写交互题需要用。

实数域二分三分，eps写法或者求精度次数

整数域三分，大范围三分，小范围暴力，