

C++编程

二分查找

binary search

快快编程
kkcoding.net

商场防盗磁门



你买了8袋商品正离开商场

突然，防盗磁门警报响了

保安想知道哪一袋有未消磁商品，要你重新过几遍磁门

请问至少要过几次才能保证知道哪个袋子有问题？

每次都将袋子平均分为两组

老师演示 "猜数字游戏"

思考程序的步骤



快快编程
kkcoding.net

老师演示 "猜数字机器人"

思考程序的步骤

猜数字机器人

```
5 int l=0,r=1000,x;
6 for(int i=1;i<=10;i++){
7     cout<<"Is it ";
8     int mid=l+(r-1)/2;
9     cout<<mid<<"?"<<endl;
10    cout<<"Too small[0], too big[1], right answer[2]:";
11    cin>>x;
12    if(x==0) l=mid+1;
13    else if(x==1) r=mid-1;
14    else break;
15 }
```

可能溢出

用 $mid=(l+r)/2$ 有问题吗?

可行解范围初始化:

$[0, 1000]$

$l=0, r=1000$

不断尝试猜中间数 $mid=l+(r-1)/2$

如果 mid 猜的太小, 设置 $l=mid+1$

如果 mid 猜的太大, 设置 $r=mid-1$

如果猜中就退出

二分查找思想

不断将可行解的范围**一分为二**：
其中的一半范围内不再查找
另外一半范围内继续查找

当原范围大小为 M 时
二分查找算法的时间复杂度为 $O(\log M)$

$\log M$ 代表2的几次方是 M

例如 $M=10^3$, 查找次数约10次

例如 $M=10^6$, 查找次数约20次

例如 $M=10^9$, 查找次数约30次

二分查找位置

C++自帶功能

快快编程
kkcoding.net

查找位置

保证序列元素**从小到大**的前提下
若插入某给定元素，可以插入哪里呢？

| | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|
| 编号 <i>i</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值 <i>x[i]</i> | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

6可以插入哪里？

插几号数之前？

可插入位置的范围{4,5,6,7}号

范围{4,5,6,7}的下边界记作

lower_bound

范围{4,5,6,7}的上边界记作

upper_bound

查找位置

保证序列元素**从小到大**的前提下
若插入某给定元素，可以插入哪里呢？

| | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|
| 编号 <i>i</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值 <i>x[i]</i> | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

8可以插入哪里？

插几号数之前？

可插入位置的范围{7,8,9}号

范围{7,8,9}号的下边界记作

lower_bound

范围{7,8,9}号的上边界记作

upper_bound

查找位置

保证序列元素**从小到大**的前提下
若插入某给定元素，可以插入哪里呢？

| | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|
| 编号 <i>i</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值 <i>x[i]</i> | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

5可以插入哪里？

插几号数之前？

可插入位置的范围{4}号

范围{4}号的下边界记作

lower_bound

范围{4}号的上边界记作

upper_bound

因为5不存在，所以上下边界重叠

查找位置

保证序列元素**从小到大**的前提下
若插入某给定元素，可以插入哪里呢？

| | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|
| 编号 <i>i</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值 <i>x[i]</i> | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

-3可以插入哪里？

插几号数之前？

可插入位置的范围{0}号

范围{0}号的下边界记作

lower_bound

范围{0}号的上边界记作

upper_bound

因为-3不存在，所以上下边界重叠

查找位置

保证序列元素**从小到大**的前提下
若插入某给定元素，可以插入哪里呢？

| | | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|--|
| 编号 <i>i</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| 数值 <i>x[i]</i> | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 | |

15可以插入哪里？

插几号数之前？

可插入位置的范围{10}号

范围{10}号的下边界记作

lower_bound

范围{10}号的上边界记作

upper_bound

因为15不存在，所以上下边界重叠

```

1  #include<iostream>
2  #include<algorithm> ←
3  using namespace std;
4  int a[10]={0,1,2,3,6,6,6,8,8,9};
5  int main(){
6      cout<<a<<endl;
7      cout<<a+1<<endl;
8      cout<<a+2<<endl;
9      cout<<a+3<<endl;
10     cout<<a+4<<endl;
11     cout<<lower_bound(a,a+10,6)<<endl;
12     int k=lower_bound(a,a+10,6)-a;
13     cout<<k<<endl;
14     cout<<a[k]<<endl;
15     cout<<*(a+k)<<endl;
16     cout<<*lower_bound(a,a+10,6)<<endl;
17     return 0;
18 }

```

lower_bound

哪行是地址(也叫指针)

哪行是数组编号

哪行是数值大小

0x472020

0x472024

0x472028

0x47202c

0x472030

0x472030

4

6

6

6

```

1  #include<iostream>
2  #include<algorithm> ←
3  using namespace std;
4  int k,a[10]={0,1,2,3,6,6,6,8,8,9};
5  int main(){
6      cout<<a<<endl;
7      cout<<a+7<<endl;
8      cout<<upper_bound(a,a+10,6)<<endl;
9      k=upper_bound(a,a+10,6)-a;
10     cout<<k<<endl;
11     cout<<a[k]<<endl;
12     cout<<*(a+k)<<endl;
13     cout<<*upper_bound(a,a+10,6)<<endl;
14     return 0;
15 }

```

upper_bound

哪行是地址(也叫指针)

哪行是数组编号

哪行是数值大小

0x472020

0x47203c

0x47203c

4

6

6

6

小结

`lower_bound()`和`upper_bound()`返回值的类型
不是`int`类型

而是指针类型（记录内存地址）

两个地址相减可以得到两者间隔，例如：

`a`是指针类型（内存地址）

`a+4` 也是指针类型（内存地址）

`(a+4) - a` 结果是`int`类型

`lower_bound(a, a+10, 6) - a` 结果是`int`类型

二分查找位置

保证序列元素**从小到大**的前提下
若插入某给定元素，可以插入哪里呢？

| | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|
| 编号 <i>i</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值 <i>x[i]</i> | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

数组起始位置*x*

`lower_bound(x, x+10, 6)`

`upper_bound(x, x+10, 6)`

6可以
插入
哪里？

二分查找位置

保证序列元素**从小到大**的前提下
若插入某给定元素，可以插入哪里呢？

| | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|
| 编号 <i>i</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值 <i>x[i]</i> | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

6可以插入的
最小编号是几号

$\text{lower_bound}(x, x+10, 6) - x$

6可以插入的
最大编号是几号

$\text{upper_bound}(x, x+10, 6) - x$

6可以
插入
哪里？

二分查找位置

从地址 x 开始到地址 $x+n$
二分查找 v 可以插入的位置最早在哪

| | | | |
|----|--------------------------------------|----------|----------|
| 地址 | $\text{lower_bound}(x, x+n, v)$ | | |
| 编号 | $\text{lower_bound}(x, x+n, v) - x$ | 记作 k | |
| 数值 | $*\text{lower_bound}(x, x+n, v)$ | 即 $x[k]$ | $*(x+k)$ |

从地址 x 开始到地址 $x+n$
二分查找 v 可以插入的位置最后在哪

| | | | |
|----|--------------------------------------|----------|----------|
| 地址 | $\text{upper_bound}(x, x+n, v)$ | | |
| 编号 | $\text{upper_bound}(x, x+n, v) - x$ | 记作 k | |
| 数值 | $*\text{upper_bound}(x, x+n, v)$ | 即 $x[k]$ | $*(x+k)$ |

存在性判断1

对于一个有序数组，要判断v是否存在，可以利用二分查找

```
1 #include<iostream>
2 #include<algorithm>
3 using namespace std;
4 int v,x[10]={0,1,2,3,6,6,6,8,8,9};
5 int main(){
6     cin>>v;
7     int lb=lower_bound(x,x+10,v)-x;
8     if(lb<10&& x[lb]==v)
9         cout<<"Yes"<<endl;
10    else
11        cout<<"No"<<endl;
12    return 0;
13 }
```

从小到大
有序排列

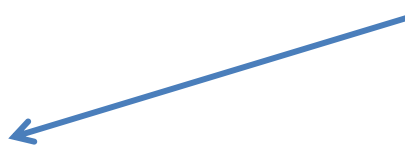
请翻译这个条件

存在性判断2

对于一个有序数组，要判断v是否存在，可以利用二分查找

```
1  #include<iostream>
2  #include<algorithm>
3  using namespace std;
4  int v,x[10]={0,1,2,3,6,6,6,8,8,9};
5  int main(){
6      cin>>v;
7      int lb=lower_bound(x,x+10,v)-x;
8      int ub=upper_bound(x,x+10,v)-x;
9      if(lb==ub)cout<<"No"<<endl;
10     else cout<<"Yes"<<endl;
11     return 0;
12 }
```

从小到大
有序排列



时间复杂度

`lower_bound(x, x+N, v)`

`upper_bound(x, x+N, v)`

均使用二分查找算法
时间复杂度均为 $O(\log N)$

保证数组
从小到大

有几个元素等于v?

| | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|
| 编号i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值x[i] | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

`lower_bound(x, x+10, 6)`

`upper_bound(x, x+10, 6)`

6可以插
入哪里?

共3个数等于6

保证数组
从小到大

有几个元素等于v?

```
1 #include<iostream>
2 #include<algorithm>
3 using namespace std;
4 int v,x[10]={0,1,2,3,6,6,6,8,8,9};
5 int main(){
6     cin>>v;
7     int lb=lower_bound(x,x+10,v)-x;
8     int ub=upper_bound(x,x+10,v)-x;
9     cout<<ub-lb<<endl;
10    return 0;
11 }
```

保证数组
从小到大

有几个元素小于等于v?

| | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|
| 编号i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值x[i] | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

数组起始位置x

`upper_bound(x, x+10, 6)`

6可以插
入哪里?

共7个数小于等于6

保证数组
从小到大

有几个元素小于等于v?

```
1 #include<iostream>
2 #include<algorithm>
3 using namespace std;
4 int v,x[10]={0,1,2,3,6,6,6,8,8,9};
5 int main(){
6     cin>>v;
7     int ub=upper_bound(x,x+10,v)-x;
8     cout<<ub<<endl;
9     return 0;
10 }
```

保证数组
从小到大

有几个元素小于v?

| | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|
| 编号i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值x[i] | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

数组起始位置x

`lower_bound(x, x+10, 6)`

6可以插
入哪里?

共4个数小于6

保证数组
从小到大

有几个元素小于v?

```
1 #include<iostream>
2 #include<algorithm>
3 using namespace std;
4 int v,x[10]={0,1,2,3,6,6,6,8,8,9};
5 int main(){
6     cin>>v;
7     int lb=lower_bound(x,x+10,v)-x;
8     cout<<lb<<endl;
9     return 0;
10 }
```

大于等于a
小于等于b

`upper_bound(x, x+10, 1)`

`lower_bound(x, x+10, 1)`

| | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|
| 编号i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值x[i] | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

大于等于1
且小于等于8
的数有几个?

`lower_bound(x, x+10, 8)`

`upper_bound(x, x+10, 8)`

`upper_bound(x, x+10, 8)` — `lower_bound(x, x+10, 1)`

大于a
小于b

upper_bound(x, x+10, 1)

lower_bound(x, x+10, 1)

| | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|
| 编号i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值x[i] | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

大于1
且小于8
的数有几个?

lower_bound(x, x+10, 8)

upper_bound(x, x+10, 8)

lower_bound(x, x+10, 8) — upper_bound(x, x+10, 1)

大于等于a
小于b

`upper_bound(x, x+10, 1)`

`lower_bound(x, x+10, 1)`

| | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|
| 编号i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值x[i] | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

大于等于1
且小于8
的数有几个?

`lower_bound(x, x+10, 8)`

`upper_bound(x, x+10, 8)`

`lower_bound(x, x+10, 8)` — `lower_bound(x, x+10, 1)`

大于a
小于等于b

`upper_bound(x, x+10, 1)`

`lower_bound(x, x+10, 1)`

| | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|
| 编号i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 数值x[i] | 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |

大于1
且小于等于8
的数有几个?

`lower_bound(x, x+10, 8)`

`upper_bound(x, x+10, 8)`

`upper_bound(x, x+10, 8)`

`upper_bound(x, x+10, 1)`

枪支购买



你是神枪手，需要为自己配备一支新的手枪。你的预算共 r 元，目前共有 n 种手枪可以选择，价格从小到大排列，其中 i 号手枪售价 $p[i]$ 元。在你的预算范围内，共有几支手枪可以考虑？

输入第一行正整数 r 和 n ，代表预算和手枪数量，均不超过1000。第二行为 n 个正整数价格均不超过10000。输出一个整数。

输入样例：

1000 3

500 900 1200



输出样例：

2

如何利用二分查找

枪支购买



```
1 #include<bits/stdc++.h>
2 using namespace std;
3 const int N=1009;
4 int p[N],n,r;
5 int main() {
6     cin>>r>>n;
7     for(int i=0;i<n;i++)cin>>p[i];
8     int ans=
9     cout<<ans<<endl;
10    return 0;
11 }
```

补全
程序

现场挑战 快快编程426

快快编程
kkcoding.net

算法步骤

将手枪价格 y 数组从小到大排序

枚举每一种狙击枪编号 i

若购买 i 号狙击枪花 $x[i]$ 元
剩 $m - x[i]$ 元

用二分查找求出
剩余资金可以考虑几种手枪
该数量累计到答案里




```
8  cin>>m>>a>>b;
9  for(int i=0;i<a;i++)cin>>x[i];
10 for(int i=0;i<b;i++)cin>>y[i];
11 
12 int cnt=0;
13 for(int i=0;i<a;i++){
14     int r=m-x[i];
15     cnt+=
16 }
17 cout<<<<endl;
```



参考资料

http://www.cplusplus.com/reference/algorithm/lower_bound/
http://www.cplusplus.com/reference/algorithm/upper_bound/

快快编程
kkcoding.net

快快编程作业

750

425

426

拓展题

751, 1662

作业要求

翻译每一行写注释
解释每个变量含义