

CS102

A Mars rover, likely a Curiosity rover, is shown on a rocky, orange-hued landscape. The rover is positioned in the center-left of the frame, facing right. It has six large, treaded wheels and a complex body with various instruments and cameras. The background shows a hazy, orange sky and distant, low hills. The overall color palette is dominated by shades of yellow and orange.

C++
算法

二分查找

binary search

猜数字游戏

请打开程序
“猜数字游戏”

请运行程序

思考程序的步骤



猜数字机器人

请打开程序
“猜数字机器人”

请运行程序

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-l)/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 }
```

可以用 $\text{mid}=(l+r)/2$ 吗?

可行解范围初始化:

$[0,1000]$
 $l=0, r=1000$

不断尝试猜中间数 $\text{mid}=l+(r-l)/2$

如果猜的太小, 设置 $l=\text{mid}+1$

如果猜的太大, 设置 $r=\text{mid}-1$

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 $O(\log M)$

例如 $M=10^6$, 查找次数约20次

二分查找1: lower_bound()

请打开程序
“lower_bound”

请运行程序

思考程序的功能

lower bound
中文意思为下边界
这里指的是
什么的下边界呢？


```
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+1<<endl;
8     cout<<a+4<<endl;
9     cout<<lower_bound(a,a+10,6)<<endl;
10    k=lower_bound(a,a+10,6)-a;
11    cout<<k<<endl;
12    k=lower_bound(a,a+10,8)-a;
13    cout<<k<<endl;
14    k=lower_bound(a,a+10,9)-a;
15    cout<<k<<endl;
16    k=lower_bound(a+5,a+10,20)-a;
17    cout<<k<<endl;
18    return 0;
19 }
```


二分查找2: upper_bound()

请打开程序
“upper_bound”

请运行程序

思考程序的功能

upper bound
中文意思为上边界
这里指的是
什么的上边界呢？

```
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+1<<endl;
8     cout<<a+7<<endl;
9     cout<<upper_bound(a,a+10,6)<<endl;
10    k=upper_bound(a,a+10,6)-a;
11    cout<<k<<endl;
12    k=upper_bound(a,a+10,8)-a;
13    cout<<k<<endl;
14    k=upper_bound(a,a+10,9)-a;
15    cout<<k<<endl;
16    k=upper_bound(a+5,a+10,20)-a;
17    cout<<k<<endl;
18    return 0;
19 }
```

规律小结

保证序列从小到大的顺序不变的前提下
若要插入某个给定元素，可以插入哪里呢？

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|

6可以插入哪里？

lower_bound
返回位置

upper_bound
返回位置

可插入
位置的
下边界

可插入
位置的
上边界

规律小结

保证序列从小到大的顺序不变的前提下
若要插入某个给定元素val

可插入
位置的下
边界

lower_bound(start,end,val)返回值
从start位置开始到end位置之前
查找能插入val元素的**最早位置**

可插入
位置的上
边界

upper_bound(start,end,val)返回值
从start位置开始到end位置之前
查找能插入val元素的**最后位置**

规律小结

`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

有序数组：存在性判断

对于一个有序数组，要判断 x 是否存在，可以利用二分查找

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

有序数组：个数统计

对于一个有序数组，要判断 x 有几个，可以利用二分查找

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 6 | 6 | 6 | 8 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|

6可以插入哪里？

lower_bound
返回位置

upper_bound
返回位置

有序数组：个数统计

对于一个有序数组，要判断x有几个，可以利用二分查找

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

另类解释

`lower_bound(start, end, val)`返回值
从start位置开始到end位置之前
查找**大于等于val**元素的最早位置
找不到时返回end

`upper_bound(start, end, val)`返回值
从start位置开始到end位置之前
查找**大于val**元素的最早位置
找不到时返回end

二分查找框架

浮点数答案

例题：开二次方根

输入正数 a ，输出 \sqrt{a} ，保留两位小数。 $a \leq 1000$

样例输入

4

样例输出

2.00

样例输入

10

样例输出

3.16

样例输入

66.66

样例输出

8.16

思考1：

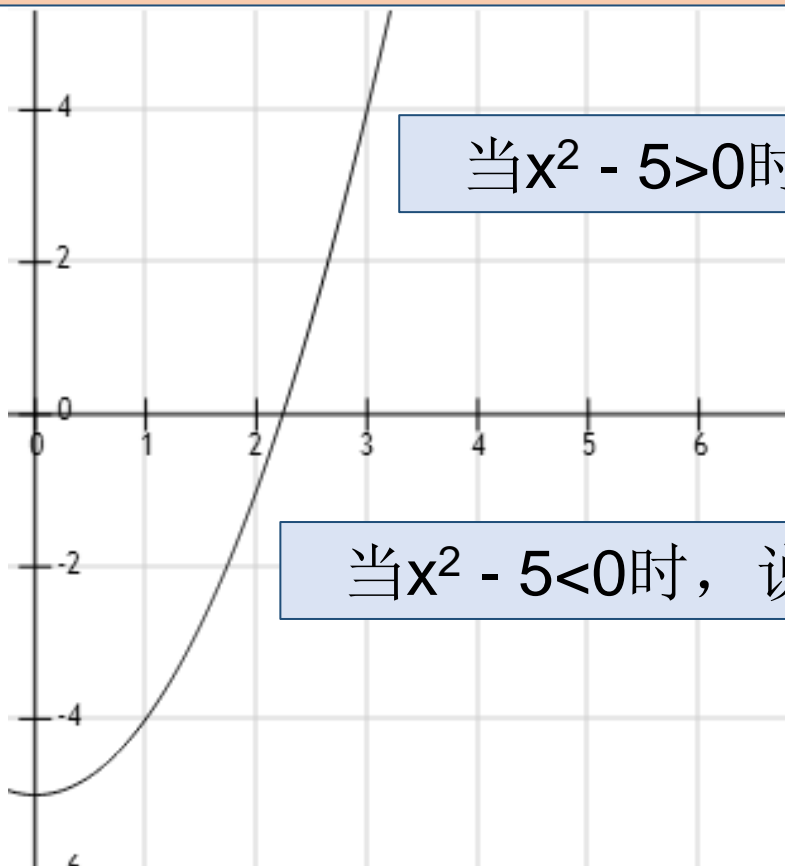
如何使用二分查找算法
计算开二次方根？

思考2：

如何确保两位小数的精确度？

例题：开二次方根

如何计算 $\sqrt{5}$ 的数值，保留两位小数



当 $x^2 - 5 > 0$ 时，说明 x 太大了

当 $x^2 - 5 < 0$ 时，说明 x 太小了

$f(x) = x^2 - 5$ 的函数图像
与 x 正半轴有唯一相交点，其横坐标就是 $\sqrt{5}$

```
1 #include<iostream>
2 #include<cmath>
3 #include<iomanip>
4 #define ERR 0.000001
5 using namespace std;
```

设置误差范围

```
6 double a;
7 bool tooSmall(double x){return x*x<a;}
```

```
8 int main() {
```

```
9     cin>>a;
```

```
10     double l=0,r=1000;
```

```
11     while(r-l>ERR){
```

```
12         double mid=l+(r-l)/2;
```

```
13         if(tooSmall(mid)) l=mid;
```

```
14         else r=mid;
```

```
15     }
```

```
16     cout<<fixed<<setprecision(2)<<r<<endl;
```

```
17     cout<<fixed<<setprecision(2)<<sqrt(a)<<endl;
```

```
18     return 0;
```

```
19 }
```

二分查找框架

参考资料

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

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

http://www.cplusplus.com/reference/algorithm/binary_search/

课件下载链接:

链接: <https://pan.baidu.com/s/1ei7f7w>

密码: q66i

作业网站:

<http://120.132.18.213:8080/thrall-web/main#home>