

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

找数组内最大和最小元素

`min_element()`

返回最小元素的位置

`max_element()`

返回最大元素的位置

如有多个一样大
返回首个位置

找数组内最大和最小元素

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

位置

下标

数值

nth_element()功能

自学资料

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

二分查找

binary search

有序数组：统计元素个数1

对于一个有序数组，要判断小于x的元素有几个？

请预测输出结果

```
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<<lower_bound(a,a+10,0)-a<<endl;
7      cout<<lower_bound(a,a+10,1)-a<<endl;
8      cout<<lower_bound(a,a+10,2)-a<<endl;
9      cout<<lower_bound(a,a+10,3)-a<<endl;
10     cout<<lower_bound(a,a+10,6)-a<<endl;
11     cout<<lower_bound(a,a+10,8)-a<<endl;
12     cout<<lower_bound(a,a+10,9)-a<<endl;
13     cout<<lower_bound(a,a+10,20)-a<<endl;
14     return 0;
15 }
```

x分别为0,1,2,3,6,6,6,8,8,9

有序数组：统计元素个数2

对于一个有序数组，要判断小于等于x的元素有几个？

```
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<<upper_bound(a,a+10,0)-a<<endl;
7      cout<<upper_bound(a,a+10,1)-a<<endl;
8      cout<<upper_bound(a,a+10,2)-a<<endl;
9      cout<<upper_bound(a,a+10,3)-a<<endl;
10     cout<<upper_bound(a,a+10,6)-a<<endl;
11     cout<<upper_bound(a,a+10,8)-a<<endl;
12     cout<<upper_bound(a,a+10,9)-a<<endl;
13     cout<<upper_bound(a,a+10,20)-a<<endl;
14     return 0;
15 }
```

请预测输出结果

x分别为0,1,2,3,6,6,6,8,8,9

浮点数危机 之 二分查找

```
1 #include<iostream>
2 #include<algorithm>
3 #define N 2
4 using namespace std;
5 double f[N]={-7.2,-7.2};
6 int main(){
7     cout<<lower_bound(f,f+N,-7.2)-f<<endl;
8     cout<<lower_bound(f,f+N,5-12.2)-f<<endl;
9     cout<<upper_bound(f,f+N,-7.2)-f<<endl;
10    cout<<upper_bound(f,f+N,5-12.2)-f<<endl;
11    return 0;
12 }
```

请预测输出结果

僵尸幸存者1

有 n 头僵尸站成一排，从左到右排列:第 i 个僵尸的坐标是 x_i 。你有一个炸弹扔在了坐标 y ，它的毁灭半径是 r 。求还剩几头僵尸幸存。输入第一行为正整数 n ，浮点数 y 和浮点数 r ，第二行为 n 个浮点数 x_i 。 $n \leq 100000$ ， $|x_i| \leq 100$ ，输出一个整数。每个浮点数最多两位小数。

输入样例:

2 5.0 12.2

-7.2 17.2

输出样例:

0

输入样例:

5 -0.2 1.0

-1.2 0.8 5.0 5.0 5.0

输出样例:

3

思考：如何使用二分查找算法

僵尸幸存者1 错误程序

```
1  #include<iostream>
2  #include<algorithm>
3  #define N 100005
4  using namespace std;
5  int n;
6  double y,r,x[N];
7  int main(){
8      cin>>n>>y>>r;
9      for(int i=0;i<n;i++) cin>>x[i];
10     int lb=lower_bound(x,x+n,y-r)-x;
11     int ub=upper_bound(x,x+n,y+r)-x;
12     int death=ub-lb;
13     cout<<n-death<<endl;
14     //cout<<lb<<endl;
15     //cout<<ub<<endl;
16     return 0;
17 }
```

思考：哪一句出现了错误？

僵尸幸存者1

```
1 #include<iostream>
2 #include<algorithm>
3 #define N 100005
4 #define ERR 0.000001
5 using namespace std;
6 bool cmp(const double&a,const double&b){
7     return a+ERR<b;
8 }
9 int n;
10 double y,r,x[N];
11 int main(){
12     cin>>n>>y>>r;
13     for(int i=0;i<n;i++) cin>>x[i];
14     int lb=lower_bound(x,x+n,y-r,cmp)-x;
15     int ub=upper_bound(x,x+n,y+r,cmp)-x;
16     int death=ub-lb;
17     cout<<n-death<<endl;
18     //cout<<lb<<endl;
19     //cout<<ub<<endl;
20     return 0;
21 }
```

返回 $a+ERR$ 是否小于 b

用于判断 a 是否排在 b 前

若 $cmp(a,b)$ 和 $cmp(b,a)$ 都返回0, 认为 a,b 相等

二分查找框架

整数答案

臭皮匠1

共 n 个臭皮匠坐成一排，从左数第 i 个人智商为 x_i ，他们想去分组挑战聪明的诸葛亮。每一组只可以安排相邻就坐的若干个臭皮匠上场挑战。为了让每一组都有资格挑战诸葛亮，必须保证每一组的智商和不低于 m ，**求最多派出几组臭皮匠？**
输入第一行为正整数 n 和 m ，第二行为 n 个正整数 x_i 。
 $n \leq 100$ ， $m \leq 1000$ ， $x_i \leq 500$ ，输出一个正整数。

输入样例：

5 100

50 100 60 60 110

输出样例：

3

输入样例：

5 100

20 10 10 30 29

输出样例：

0

臭皮匠1：贪心算法

输入样例：

5 100

50 100 60 60 110

输出样例：

3

50

100

60

60

110

贪心算法：从左向右，依次安排每个臭皮匠加入当前小组。当前小组如果凑满 m 的智商，就新开一组

```

1  #include<iostream>
2  #include<algorithm>
3  #define N 105
4  using namespace std;
5  int i,n,m,x[N];
6  int main(){
7      cin>>n>>m;
8      for(i=0;i<n;i++) cin>>x[i];
9      int cnt=0,sum=0;
10     for(i=0;i<n;i++){
11         sum+=x[i];
12         if(sum>=m){
13             sum=0;
14             cnt++;
15         }
16     }
17     cout<<cnt<<endl;
18     return 0;
19 }

```

cnt记录已有几个
小组完成智商和
不低于m的要求

sum记录当前该小
组成员智商和

臭皮匠2

共 n 个臭皮匠坐成一排，第 i 个人智商为 x_i ，他们想去分组挑战聪明的诸葛亮。每一组只可以安排相邻就坐的若干个臭皮匠上场挑战。现在必须组织起 k 次挑战，为了不要让围观者看不起臭皮匠团体，**希望使各组臭皮匠智商和的最小值越大越好，请问这个数最大是多少？**

输入第一行为正整数 n 和 k ，第二行为 n 个正整数 x_i 。
 $1 \leq k \leq n \leq 100$ ， $x_i \leq 500$ ，输出一个正整数。

输入样例：

5 3

50 100 60 60 110

输出样例：

110

输入样例：

3 2

20 20 30

输出样例：

30

思考：和臭皮匠1有啥关系

思考：如何使用二分查找算法

臭皮匠2与臭皮匠1的关系

臭皮匠2算法：暴力枚举答案

从小到大枚举可能的答案: $m=0,1,2,3,\dots,50000$

对于特定的 m ,利用臭皮匠1的算法判断:保证每一组的智商和不低于 m 时,能否派出 k 组臭皮匠?

如果发现对于某个 m 无法派出 k 组臭皮匠,就停止枚举,输出答案 $m-1$

时间复杂度 $O(M*N)$

M 是答案可能的范围大小

N 是臭皮匠人数的上限

臭皮匠2： 二分答案+判断可行性

臭皮匠2算法： 二分枚举答案

二分枚举可能的答案: m 范围初始化为 $[0, 50000]$ 中整数

对于特定的 m ,利用臭皮匠1的算法判断:保证每一组的智商和不低于 m 时, 能否派出 k 组臭皮匠?

不断二分缩小 m 的可能范围, 直到待查找范围为空

时间复杂度 $O(N \cdot \log M)$
 M 是答案可能的范围大小
 N 是臭皮匠人数的上限

```

6  bool OK(int m){//让每组至少m时, 能否有k组
7      int cnt=0, sum=0;
8      for(i=0; i<n; i++)
9          if((sum+=x[i])>=m) sum=0, cnt++;
10     return cnt>=k;
11 }

```

```

15     int l=*min_element(x, x+n);
16     int r=0;
17     for(i=0; i<n; i++) r+=x[i];
18     int ans=l;
19     while(l<=r){
20         int mid=l+(r-l)/2;
21         if(OK(mid)) ans=mid, l=mid+1;
22         else r=mid-1;
23     }
24     cout<<ans<<endl;

```

初始化左端点:最低智商

初始化右端点:智商总和

初始化答案

当还存在待查找范围

二分范围, 中点为mid

若mid是可行解, 就找更大可行解;
否则, 就找更小可行解

小结

找最优解问题



在答案范围内二分查找



判断可行性问题

二分查找框架： 整数答案

```
int l= 初始化左端点
int r= 初始化右端点
int ans= 初始化答案
while(l<=r){ 当还存在待查找范围
    int mid=l+(r-l)/2; 二分范围：中点为mid
    if(OK(mid)) 根据mid是否为可行解
    else 更新范围端点和答案
}
cout<<ans<<endl;
```

易错点汇总

请打开程序
“易错点汇总”

请找出所有错误

修改程序直到全部正确

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>