

# 思想方法

## 二分

```
1 //small to big
2 #include<cstdio>
3 #include<iostream>
4 #include<cstring>
5 #include<algorithm>
6 using namespace std;
7 int search(int *arr,int n,int key){
8     int left=0,right=n-1;
9     while(left<=right){
10         int mid=(left+right)>>1;
11         if(arr[mid]==key) return mid;
12         else if(arr[mid]>key) right=mid-1;
13         else left=mid+1;
14     }
15     return -1;
16 }
17 int searchFirstEqual(int *arr,int n,int key){
18     int left=0,right=n-1;
19     while(left<=right){
20         int mid=(left+right)>>1;
21         if(arr[mid]>=key) right=mid-1;
22         else left=mid+1;
23     }
24     if(left<n&&arr[left]==key){
25         return left;
26     }
27     return -1;
28 }
29 int searchLastEqual(int *arr,int n,int key){
30     int left=0,right=n-1;
31     while(left<=right){
32         int mid=(left+right)>>1;
33         if(arr[mid]>key) right=mid-1;
34         else left=mid+1;
35     }
36     if(right>=0&&arr[right]==key){
37         return right;
38     }
39     return -1;
40 }
41 int searchLastEqualOrSmaller(int *arr,int n,int key){
42     int left=0,right=n-1;
43     while(left<=right){
44         int mid=(left+right)>>1;
45         if(arr[mid]>key) right=mid-1;
46         else left=mid+1;
47     }
```

```

48     return right;
49 }
50 int searchLastSmaller(int *arr,int n,int key){
51     int left=0,right=n-1;
52     while(left<=right){
53         int mid=(left+right)>>1;
54         if(arr[mid]>=key) right=mid-1;
55         else left=mid+1;
56     }
57     return right;
58 }
59 int searchFirstEqualOrLarger(int *arr,int n,int key){
60     int left=0,right=n-1;
61     while(left<=right){
62         int mid=(left+right)>>1;
63         if(arr[mid]>=key) right=mid-1;
64         else left=mid+1;
65     }
66     return left;
67 }
68 int searchFirstLarger(int *arr,int n,int key){
69     int left=0,right=n-1;
70     while(left<=right){
71         int mid=(left+right)>>1;
72         if(arr[mid]>key) right=mid-1;
73         else left=mid+1;
74     }
75     return left;
76 }
77
78 int main(){
79     int arr[17] = {1,
80                   2, 2, 5, 5, 5,
81                   5, 5, 5, 5, 5,
82                   5, 5, 6, 6, 7};
83     printf("First Equal      : - \n", searchFirstEqual(arr, 16, 5));
84     printf("Last Equal       : - \n", searchLastEqual(arr, 16, 5));
85     printf("First Equal or Larger : - \n", searchFirstEqualOrLarger(arr, 16, 5));
86     printf("First Larger      : - \n", searchFirstLarger(arr, 16, 5));
87     printf("Last Equal or Smaller : - \n", searchLastEqualOrSmaller(arr, 16, 5));
88     printf("Last Smaller       : - \n", searchLastSmaller(arr, 16, 5));
89     return 0;
90 }

```

## 三分

```

1 #include<iostream>
2 #include<cstdio>
3 using namespace std;
4 int n;
5 double a[200];
6 double l,r;
7 double speed(double a,int b){

```

```

8      double cur=a,ans=1;
9      while(b){
10         if(b&1) ans*=cur;
11         cur*=cur;
12         b>>=1;
13     }
14     return ans;
15 }
16 double num(double x){
17     double sum=0;
18     for(int i=0;i<=n;i++){
19         sum=sum+speed(x,i)*a[i];
20     }
21     return sum;
22 }
23 double abs(double c){
24     if(c<0) return -c;
25     return c;
26 }
27 int main(){
28     scanf("%d%lf%lf",&n,&l,&r);
29     for(int i=1;i<=n+1;i++){
30         scanf("%lf",&a[n-i+1]);
31     }
32     while(abs(r-l)>0.000001){
33         double x=(2*l+r)/3.0,y=(2*r+1)/3.0;
34         if(num(x)>num(y)){
35             r=y;
36         }
37         else{
38             l=x;
39         }
40     }
41     printf("%.5lf",l);
42     return 0;
43 }
44 }
```

## 离散化

```

1  //unique,lower_bound,sort,离散化
2  #include<iostream>
3  #include<cstdio>
4  #include<algorithm>
5  using namespace std;
6  const int MAXN=10020;
7  int a[MAXN],b[MAXN];
8  int n,m;
9  int main(){
10     scanf("%d",&n);
11     for(int i=1;i<=n;i++){
12         scanf("%d",&a[i]);
13         b[i]=a[i];
```

```
14     }
15     sort(b+1,b+1+n);
16     m=unique(b+1,b+1+n)-(b+1);
17     for(int i=1;i<=n;i++){
18         a[i]=lower_bound(b+1,b+1+m,a[i])-b;
19     }
20     for(int i=1;i<=n;i++){
21         cout<<a[i]<<' ';
22     }
23     return 0;
24 }
```