

查找

#顺序查找法

(1) 顺序查找的方法就是对于给定的key值，遍历线性表，找到这个值。

查找第*i*个元素，需要比较*n+1-i*次，如果查找失败，要比较*n+1*次。时间复杂度是 $O(n)$;

设每一个元素的被查找概率相同，则平均查找长度是 $1/n * (\sum (n-i+1)) = (n+1)/2$; 平均查找失败的长度是*n*; 查找成功最少次数是1次，查找成功最多次数是*n*次，查找失败次数是*n*次

(2) 顺序查找的优缺点：优点是对于数据结构都适用，算法简单
缺点是效率太低，平均查找长度很长

#二分查找/折半查找

(1) 给定key值，并任取序列中的序号折中值*x*，如果*x>key*，右指针左移；如果*x<key*，左指针右移；如果相等，查找成功并结束循环。

平均查找长度为 $\log(n+1)-1$ ，时间复杂度为 $O(\log(n))$

原题链接：

[P2249 【深基13.例1】查找 - 洛谷 | 计算机科学教育新生态 \(luogu.com.cn\)](https://www.luogu.com.cn/problem/P2249)

```
#include<bits/stdc++.h>
#include<algorithm>
typedef long long ll;
using namespace std;
ll n;
ll arr[1000005];
int find(int x)
{
    ll l=0;
    ll r=n;
    ll mid;
    while(l<r)
    {
        mid=(l+r)/2;
        if(arr[mid]>=x)
        {
            r=mid;
        }
        else
```

```

        l=mid+1;

    }
    if(arr[l]==x)
    {
        return l;
    }
    else return -1;

}

int main (void)
{
    cout.tie(0);
    cin.tie(0);

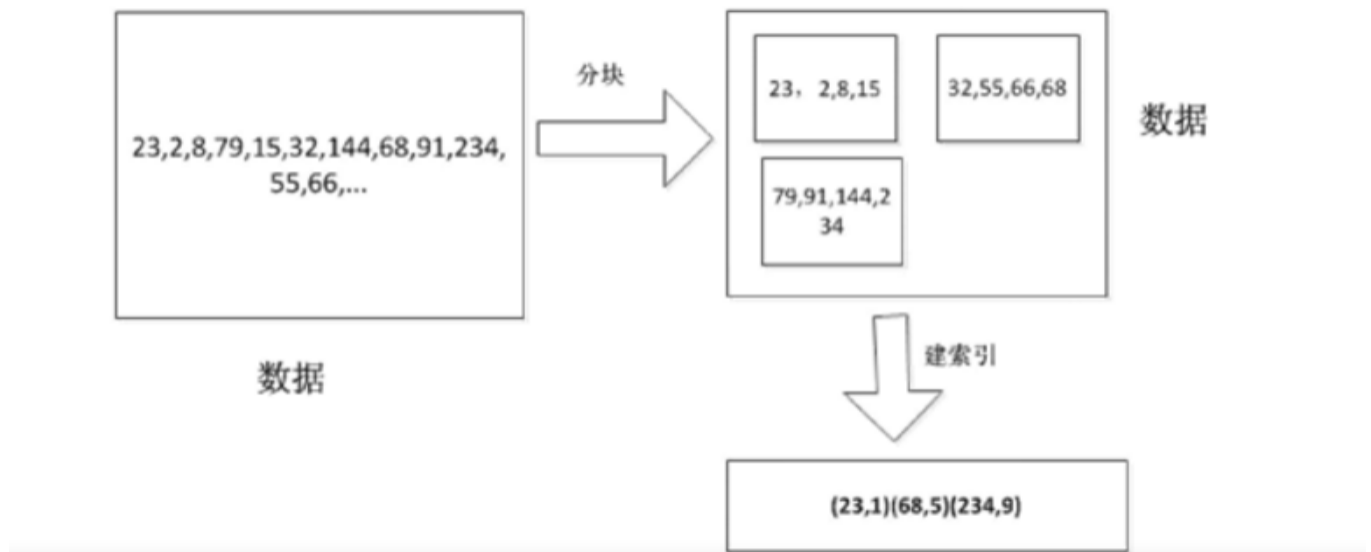
    cin>>n;
    ll m;
    cin>>m;
    for(int i=0;i<n;i++)
    {
        cin>>arr[i];
    }
    for(int i=0;i<m;i++)
    {
        int in;
        cin>>in;
        if(find(in)==-1)
            cout<<"-1"<<" ";
        else
            cout<<find(in)+1<<" ";

    }
    return 0;
}

```

#分块查找

(1) 建立索引：首先将数组分为几块：



一般为在数据中任意抽出几个元素，比前一个元素大而比后一个元素小的元素都在一个块内。索引一般为一个块内最大的元素，和这个块内对应的第一个元素在原数组中的位置（从一开始）

如：(23,1)以此类推。

(2) 查找方式：首先一一比对各组索引和所要查找目标的大小，确定了目标在哪个块后再在块中一个一个进行比较。