



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY

C++程序设计

Programming in C++



1011018

主讲：魏英，计算机学院

数组的应用

- ◆ 3、顺序查找法
- ◆ 4、二分查找法

- ▶ **顺序查找**的基本思想是让关键字与序列中的数逐个比较，直到找出与给定关键字相同的数为止或序列结束，一般应用于无序序列查找。

16.3 顺序查找法

【例16.3】顺序查找

```
1 #include <iostream>
2 using namespace std;
3 int Search(int A[],int n,int find)
4 { //顺序查找 n=序列元素个数 find=欲查找数据
5     int i;
6     for (i=0; i<n ; i++) if (A[i]==find) return i;
7     return -1; //未找到
8 }
9 #define N 10
10 int main()
11 {
12     int A[N]={18,-3,-12,34,101,211,12,90,77,45}, i,find;
13     cin>>find;
14     i=Search(A,N,find);
15     if(i>=0) cout<<"A["<<i<<"]="<<find<<endl;
16     else cout<<"not found"<<endl;
17     return 0;
18 }
```

- ▶ 对于有序序列，可以采用二分查找法进行查找。它的基本思想是：升序排列的 n 个元素集合 A 分成个数大致相同的两部分，取 $A[n/2]$ 与欲查找的 $find$ 作比较，如果相等则表示找到 $find$ ，算法终止。如果 $find < A[n/2]$ ，则在 A 的前半部继续搜索 $find$ ，如果 $find > A[n/2]$ ，则在 A 的后半部继续搜索 $find$ 。

16.4 二分查找法

【例16.4】二分查找

```
1  #include <iostream>
2  using namespace std;
3  int BinarySearch(int A[],int n,int find)
4  { //二分查找 n=序列元素个数 find=欲查找数据
5      int low,upper,mid;
6      low=0 , upper=n-1; //左右两部分
7      while(low<=upper) {
8          mid = low + (upper-low)/2;
9          //不用(upper+low)/2, 避免upper+low溢出
10         if( A[mid] < find) low = mid+1; //右半部分
11         else if (A[mid] > find) upper = mid - 1; //左半部分
12         else return mid; //找到
13     }
14     return -1; //未找到
15 }
```

16.4 二分查找法

```
15 #define N 10
16 int main()
17 {
18     int A[N]={8,24,30,47,62,68,83,90,92,95},i,find;
19     cin>>find;
20     i=BinarySearch(A,N,find);
21     if(i >= 0) cout<<"A["<<i<<"]="<<find<<endl;
22     else cout<<"not found"<<endl;
23     return 0;
24 }
```

CP 程序设计