MA144: Problem Solving and Computer Programming

Lecture-22

Sorting and Searching

Sorting

Sorting

The Basic Problem

- Given an array A[0], A[1], ... , A[size-1], reorder the entries such that
  A[0] <= A[1] <= . . . <= A[size-1]
  (List is in non-decreasing order)

Insertion Sort

Basic Idea

- The insertion sort scans input array A[n] from A[0] to A[n-1]
- It inserts each element A[k] into its proper position in the previously sorted sub array A[0], A[1], …., A[k-1].

Algorithm

Input: A[n]

```
for j=1 to n-1
 {  key=A[j]
     // insert A[j] into the sorted sub array A[0]…A[j-1]
     i=j-1
    while (i>=0 and A[i]>key)
     { A[i+1]=A[i]
        i=i-1
      }
  A[i+1]=key
 }
```

```
#include<iostream>
using namespace std;
int SIZE=50;
int main()
{  int a[SIZE], i,n,j,key,k;
   cout<<" enter number of arrary elements: ";
   cin>>n;
   for(k=0;k<n;k++)
    { cout<<"enter "<<k+1<<" element: ";
      cin>>a[k];
      cout<<endl;
    }
    for(j=1;j<n;j++)
    {
       key=a[j];
       i=j-1;
       while(i>=0 && a[i]>key)
       { a[i+1]=a[i];
          i=i-1;
        }
     a[i+1]=key;
    }
    cout<<"the sorted array is\n";
    for(k=0;k<n;k++)
      cout<<a[k]<<' ';
 return 0;
}
```

```
 enter number of elements to find an average: 7
enter 1 element: -1

enter 2 element: 4

enter 3 element: 7

enter 4 element: 2

enter 5 element: 3

enter 6 element: 8

enter 7 element: -6

the sorted array is
-6 -1 2 3 4 7 8
--------------------------------
```

Printing all iterations

```
#include<iostream>
using namespace std;
int SIZE=50;
int main()
{  int a[SIZE], i,n,j,key,k;
   cout<<" enter number of array elements: ";
   cin>>n;
   for(k=0;k<n;k++)
    { cout<<"enter "<<k+1<<" element: ";
      cin>>a[k];
      cout<<endl;
    }
     for(j=1;j<n;j++)
     {
        key=a[j];
        i=j-1;
        while(i>=0 && a[i]>key)
        { a[i+1]=a[i];
           i=i-1;
         }
      a[i+1]=key;
     cout<<j<<"th iteration:  ";
     for(k=0;k<n;k++)
      cout<<a[k]<<' ';
      cout<<endl;
     }
 return 0;
}
```

```
 enter number of elements to find an average: 7
enter 1 element: -1

enter 2 element: 4

enter 3 element: 7

enter 4 element: 2

enter 5 element: 3

enter 6 element: 8

enter 7 element: -6

1th iteration:  -1 4 7 2 3 8 -6
2th iteration:  -1 4 7 2 3 8 -6
3th iteration:  -1 2 4 7 3 8 -6
4th iteration:  -1 2 3 4 7 8 -6
5th iteration:  -1 2 3 4 7 8 -6
6th iteration:  -6 -1 2 3 4 7 8
```

Selection Sort

Basic Idea

Input array A[n]

- First find the smallest element of A, and exchange it with A[0]
- Find the second smallest of A, and exchange it with A[1]
- …….
- Continue this till n-2 element of A

Algorithm

Input: A[n]

```
for i=0 to n-2
 {  small=i
     // find smallest in the sub array A[i]…A[n-1]
     for j=i+1 to n-1
     { if(A[j]<A[small])
            small=j
      }
    swap(A[i], A[small])
 }
```

```
using namespace std;
int SIZE=50;
int main()
{ int a[SIZE], i, j, small, n, k,t;
  cout<<" enter number of array elements: ";
  cin>>n;
  cout<<"enter array: ";
  for(k=0;k<n;k++)
    cin>>a[k];
  for(i=0;i<n-1;i++)
  {
    small=i;
     for(j=i+1;j<n;j++)
      {  if(a[j]<a[small])
          small=j;
       }
    //swap(a[i],a[small])
     t=a[i]; a[i]=a[small]; a[small]=t;
  }
  cout<<"\n the sorted array: ";
  for(k=0;k<n;k++)
    cout<<a[k]<<' ';
return 0;
}
```

```
 enter number of array elements: 9
enter array: -1 5 3 9 12 4 8 23 4

 the sorted array: -1 3 4 4 5 8 9 12 23
--------------------------------
```

Printing all iterations

```
#include <iostream>
using namespace std;
int SIZE=50;
int main()
{ int a[SIZE], i, j, small, n, k,t;
  cout<<" enter number of array elements: ";
  cin>>n;
  cout<<"enter array: ";
  for(k=0;k<n;k++)
    cin>>a[k];
   for(i=0;i<n-1;i++)
   {
     small=i;
      for(j=i+1;j<n;j++)
       {  if(a[j]<a[small])
            small=j;
        }
      //swap(a[i],a[small])
       t=a[i]; a[i]=a[small]; a[small]=t;
     cout<<"\n array in "<<i+1<<" iteration : ";
     for(k=0;k<n;k++)
     cout<<a[k]<<' ';
   }
  return 0;
}
```

```
 enter number of array elements: 9
enter array: -1 5 3 9 12 4 8 23 15

 array in 1 iteration : -1 5 3 9 12 4 8 23 15
 array in 2 iteration : -1 3 5 9 12 4 8 23 15
 array in 3 iteration : -1 3 4 9 12 5 8 23 15
 array in 4 iteration : -1 3 4 5 12 9 8 23 15
 array in 5 iteration : -1 3 4 5 8 9 12 23 15
 array in 6 iteration : -1 3 4 5 8 9 12 23 15
 array in 7 iteration : -1 3 4 5 8 9 12 23 15
 array in 8 iteration : -1 3 4 5 8 9 12 15 23
--------------------------------
```

Searching

Searching

Check if a given element (called key) occurs in the array.

Example: array of student records;

rollno can be the key.

Two methods to be discussed:

- If the array elements are unsorted.

  Linear search

- If the array elements are sorted.

  Binary search

Linear Search

Basic idea:

- Start at the beginning of the array.
- Inspect elements one by one to see if it matches the key.

Algorithm

Input: A[n], key

```
for i=0 to n-1
 {  if (key=A[i])
     {  pos=i+1
        boolean=true
        break
      }
 }
 if(boolean)
    key is found and its position is pos
 else
    key is not found
```

```
#include<iostream>
using namespace std;
int SIZE=50;
int main()
{  int a[SIZE], i,n,j,key,k,pos;
   bool t=false;
   cout<<" enter number of array elements: ";
   cin>>n;
   cout<<"enter an array: ";
   for(k=0;k<n;k++)
     cin>>a[k];
   cout<<"enter key to be searched : ";
     cin>>key;
    for(j=0;j<n;j++)
    {   if(key==a[j])
        { pos=j+1;
          t=true;
          break;
         }
    }
    if(t)
      cout<<" key found and its position is "<<pos;
    else cout<<"key not found";
 return 0;
}
```

```
 enter number of array elements: 5
enter an array: 9 -4 5 2 0
enter key to be searched : -4
 key found and its position is 2
--------------------------------
```

```
 enter number of array elements: 6
enter an array: 2 5 -1 4 0 6
enter key to be searched : 10
key not found
--------------------------------
```