

Basic Sorting Programs

CSE 225 - Data Structures and Algorithms

Md. Mahfuzur Rahman
ECE Department
North South University

1 Insertion Sort

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     int a[] = {16, 190, 11, 15, 10, 12, 14};
7     int n=7;
8
9     for(int i=1;i<n;i++)
10    {
11        int temp=a[i];
12        int j=i-1;
13
14        while((temp<a[j])&&(j>=0))
15        {
16            a[j+1]=a[j];    //moves element forward
17            j=j-1;
18        }
19        a[j+1]=temp;    //insert element in proper place
20    }
21
22    for(int i=0;i<7;i++)
23        printf("%d\t",a[i]);
24
25    return 0;
26 }
27 }
```

2 Bubble Sort

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int a[10],n;
6     printf("Enter the number of elements: ");
7     scanf("%d",&n);
8
9     printf("Enter the array elements: ");
10    for(i=0;i<n;++i)
11        scanf("%d",&a[i]);
12
13    for(int i=1;i<n;++i)
14    {
15        for(int j=0;j<(n-i);++j)
16        {
17            if(a[j]>a[j+1])
18            {
19                int temp=a[j];
20                a[j]=a[j+1];
21                a[j+1]=temp;
22            }
23        }
24    }
25
26    printf("\nArray after sorting: ");
27    for(int i=0;i<n;++i)
28        printf("%d ",a[i]);
29    return 0;
30 }
```

3 Selection Sort

```
1 #include <stdio.h>
2
3 #define n 7
4
5 int a[n] = {4,6,3,2,1,9,7};
6
7 void display() {
8     int i;
9     printf("\t");
10
11     for(i = 0; i < n; i++) { // navigate through all items
12         printf("%d ", a[i]);
13     }
14
15     printf("\n\n");
16 }
17
18 void selectionSort() {
19
20     for(int i = 0; i < n-1; i++) { // loop through all numbers
21
22         int indexMin = i; // set current element as minimum
23
24         for(int j = i+1; j < n; j++) { // check the element to be minimum
25             if(a[j] < a[indexMin]) {
26                 indexMin = j;
27             }
28         }
29         if(indexMin != i) {
30             printf("Items swapped:\t[ %d, %d ]\n", a[i], a[indexMin]);
31
32             // swap the numbers
33             int temp = a[indexMin];
34             a[indexMin] = a[i];
35             a[i] = temp;
36         }
37
38         printf("Iteration %d#:", (i+1));
39         display();
40     }
41 }
42
43 int main() {
44     printf("Input Array: ");
45     display();
46
47     selectionSort();
48     printf("\nOutput Array: ");
49     display();
50
51 }
```

4 Sortings using a Class

```
1 #include <iostream>
2 using namespace std;
3 class Sort {
4     int arr[50];
5     int n;
6
7     public:
8     void userInput();
9     void fixedInput();
10    void bubble();
11    void insertion();
12    void selection();
13    void display();
14 };
15
16 void Sort::userInput()
17 {
18     cout<<"Enter size of elements:" << endl;
19     cin>>n;
20
21     cout<<"Enter " << n << " values" << endl;
22     for(int b = 0; b <= n -1; b++) {
23         cout << "Value [" << b << "]:";
24         cin>>arr[b];
25     }
26     cout << "Elements of your unsorted list: [ ";
27     for(int c = 0; c <= n -1; c++){
28         cout << arr[c] << " ";
29     }
30     cout << "]" << endl;
31 }
32
33 void Sort::fixedInput()
34 {
35     n=4;
36     arr[0]=4;
37     arr[1]=2;
38     arr[2]=1;
39     arr[3]=3;
40 }
41
42 void Sort::bubble()
43 {
44     for(int i=1;i<n;++i) {
45         for(int j=0;j<(n-i);++j) {
46             if(arr[j]>arr[j+1]) {
47                 int temp=arr[j];
48                 arr[j]=arr[j+1];
49                 arr[j+1]=temp;
50             }
51     }
```

```

52     }
53 }
54 void Sort::insertion()
55 {
56     for(int i=1;i<n;i++)
57     {
58         int temp=arr[i];
59         int j=i-1;
60         while( (temp<arr[j]) && (j>=0) )
61         {
62             arr[j+1]=arr[j];    //moves element forward
63             j=j-1;
64         }
65         arr[j+1]=temp;    //insert element in proper place
66     }
67 }
68
69 void Sort::selection()
70 {
71     for(int i = 0; i < n-1; i++) {
72         int indexMin = i;
73         for(int j = i+1; j < n; j++) {
74             if(arr[j] < arr[indexMin]) {
75                 indexMin = j;
76             }
77         }
78
79         if(indexMin != i) {
80             // swap the numbers
81             int temp = arr[indexMin];
82             arr[indexMin] = arr[i];
83             arr[i] = temp;
84         }
85     }
86 }
87 }
88 void Sort::display()
89 {
90     cout << "Elements of your sorted list: [ ";
91     for(int x = 0; x < n; x++) {
92         cout << arr[x] << " ";
93     }
94     cout << "]" ;
95 }
96 int main () {
97     Sort mSort;
98     mSort.fixedInput();
99     //mSort.bubble();
100    mSort.insertion();
101    //mSort.selection();
102
103    mSort.display();
104    return 0;
105 }

```

5 Sortings using a Class (with Template)

```
1 #include <iostream>
2 using namespace std;
3 template <typename ItemType>
4 class Sort {
5     ItemType arr[50];
6     int n;
7
8     public:
9         void userInput();
10        void fixedInput();
11        void bubble();
12        void insertion();
13        void selection();
14        void display();
15 };
16
17 template <typename ItemType>
18 void Sort<ItemType>::userInput()
19 {
20     cout<<"Enter size of elements:" << endl;
21     cin>>n;
22
23     cout<<"Enter " << n << " values" << endl;
24     for(int b = 0; b <= n -1; b++) {
25         cout << "Value [" << b << "]:";
26         cin>>arr[b];
27     }
28
29     cout << "Elements of your unsorted list: [ ";
30     for(int c = 0; c <= n -1; c++){
31         cout << arr[c] << " ";
32     }
33     cout << "]" << endl;
34 }
35
36 template <typename ItemType>
37 void Sort<ItemType>::fixedInput()
38 {
39     n=4;
40     arr[0]=4.3;
41     arr[1]=2.2;
42     arr[2]=1.6;
43     arr[3]=3.2;
44 }
45
46 template <typename ItemType>
47 void Sort<ItemType>::bubble()
48 {
49     for(int i=1;i<n;++i)
50     {
51         for(int j=0;j<(n-i);++j)
```

```
52     {
53         if (arr[j] > arr[j+1])
54         {
55             ItemType temp = arr[j];
56             arr[j] = arr[j+1];
57             arr[j+1] = temp;
58         }
59     }
60 }
61 }
62
63 template <typename ItemType>
64 void Sort<ItemType>::insertion()
65 {
66     for(int i = 1; i < n; i++)
67     {
68         ItemType temp = arr[i];
69         int j = i - 1;
70
71         while( (temp < arr[j]) && (j >= 0) )
72         {
73             arr[j+1] = arr[j];    //moves element forward
74             j = j - 1;
75         }
76
77         arr[j+1] = temp;    //insert element in proper place
78     }
79 }
80
81 template <typename ItemType>
82 void Sort<ItemType>::selection()
83 {
84     for(int i = 0; i < n-1; i++)
85     {
86
87         int indexMin = i;
88
89         for(int j = i+1; j < n; j++) {
90             if(arr[j] < arr[indexMin]) {
91                 indexMin = j;
92             }
93         }
94
95         if(indexMin != i) {
96             // swap the numbers
97             ItemType temp = arr[indexMin];
98             arr[indexMin] = arr[i];
99             arr[i] = temp;
100         }
101     }
102 }
103 }
104
105 template <typename ItemType>
```



```
106 void Sort<ItemType>::display ()
107 {
108     cout << "Elements of your sorted list: [ ";
109     for(int x = 0; x < n; x++) {
110         cout << arr[x] << " ";
111     }
112     cout << "]" ;
113 }
114
115 int main () {
116     Sort<float> mSort;
117     mSort.fixedInput();
118     mSort.bubble();
119     //mSort.insertion();
120     //mSort.selection();
121
122     mSort.display();
123     return 0;
124 }
```

6 Sortings using a Class (with ItemType Object)

```
1 #include <iostream>
2 using namespace std;
3
4 class ItemType{
5     public:
6         int id;
7         float result;
8 };
9
10 class Sort {
11     ItemType arr[50];
12     int n;
13
14     public:
15         void userInput();
16         void fixedInput();
17         void bubble();
18         void insertion();
19         void selection();
20         void display();
21 };
22
23 void Sort::userInput()
24 {
25     cout<<"Enter size of elements:" << endl;
26     cin>>n;
27
28     cout<<"Enter " << n << " values" << endl;
29
30     for(int b = 0; b <= n -1; b++) {
31         cin>>arr[b].id;
32
33         cout << "Value [" << b << "] (result):";
34         cin>>arr[b].result;
35     }
36
37     cout << "Elements of your unsorted list: [ ";
38     for(int c = 0; c <= n -1; c++){
39         cout << "("<<arr[c].id<<","<<arr[c].result<<") " ";
40     }
41     cout << "]" << endl;
42 }
43
44 void Sort::fixedInput()
45 {
46     {
47         n=4;
48         arr[0].id=1; arr[0].result=5.5;
49         arr[1].id=2; arr[1].result=2.1;
50         arr[2].id=3; arr[2].result=3.7;
51         arr[3].id=4; arr[3].result=4.2;
```

```
52 }
53
54 void Sort::bubble()
55 {
56
57     for(int i=1;i<n;++i)
58     {
59         for(int j=0;j<(n-i);++j)
60         {
61             if(arr[j].result>arr[j+1].result)
62             {
63                 ItemType temp=arr[j];
64                 arr[j]=arr[j+1];
65                 arr[j+1]=temp;
66             }
67         }
68     }
69 }
70
71 void Sort::insertion()
72 {
73     for(int i=1;i<n;i++)
74     {
75         ItemType temp=arr[i];
76         int j=i-1;
77
78         while( (temp.result<arr[j].result) && (j>=0) )
79         {
80             arr[j+1]=arr[j];    //moves element forward
81             j=j-1;
82         }
83
84         arr[j+1]=temp;    //insert element in proper place
85     }
86 }
87
88 void Sort::selection()
89 {
90     for(int i = 0; i < n-1; i++)
91     {
92         int indexMin = i;
93
94         for(int j = i+1;j < n;j++) {
95             if(arr[j].result < arr[indexMin].result) {
96                 indexMin = j;
97             }
98         }
99
100         if(indexMin != i) {
101             // swap the numbers
102             ItemType temp = arr[indexMin];
103             arr[indexMin] = arr[i];
104             arr[i] = temp;
105         }
106     }
```

```
106
107     }
108 }
109 void Sort::display()
110 {
111     cout << "Elements of your sorted list: [ ";
112     for(int c = 0; c < n; c++) {
113         cout << "("<<arr[c].id<<","<<arr[c].result<<") ";
114     }
115     cout << " ]";
116 }
117 }
118
119 int main() {
120     Sort mSort;
121     mSort.fixedInput();
122     mSort.bubble();
123     //mSort.insertion();
124     //mSort.selection();
125
126     mSort.display();
127     return 0;
128 }
```