Lecture – 15

Shorting: Sometimes we need to short out a list of elements by ascending order or descending order. For this purpose we can apply various shorting algorithms.

1. Insertion sort
2. Bubble sort
3. Merge sort
4. Quick sort
5. Radix sort
6. Heap sort
7. Selection sort

Insertion sort:

```
void main()
{
        int a[50], i, j, n, t;
        scanf("%d",&n);

        for( i = 0; i < n; i ++)
                scanf("%d", & a[i]);

        for( j = 1; j < n; j ++){
                i = j-1;
                t = a[ j ];
                while( (i > = 0)&&(a[i] > t)){
                                a[i+1] = a[i];
                                i -- ;
                }
                a[i+1] = t ;
        }

        for( i = 0; i < n; i ++)
                printf("%d", a[i]);
}
```

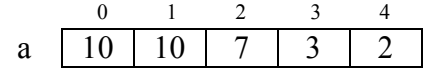
→ Shorting logic

Example: Suppose that we have a list of 5 elements. We will use the insertion sort algorithm to sort the given list in ascending order.

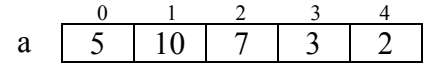
	0	1	2	3	4
a	10	5	7	3	2

Pass one: j = 1, i = 0, t = a[1] = 5

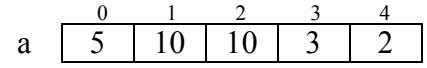
(0 > = 0) && (a[0] > 5) ⟶ TRUE



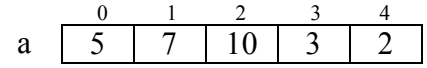
i = -1, t = 5
(-1 > = 0) && (a[-1] > 5) ⟶ FALSE



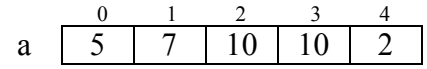
Pass two: j = 2, i = 1, t = a[2] = 7
(1 > = 0) && (a[1] > 7) ⟶ TRUE



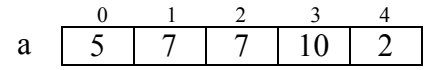
i = 0, t = 7
(0 > = 0) && (a[0] > 7) ⟶ FALSE



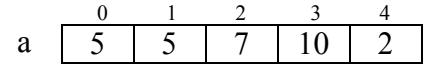
Pass three: j = 3, i = 2, t = a[3] = 3
(2 > = 0) && (a[2] > 3) ⟶ TRUE



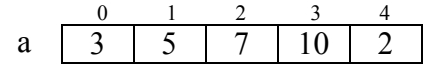
i = 1, t = 3
(1 > = 0) && (a[1] > 3) ⟶ TRUE



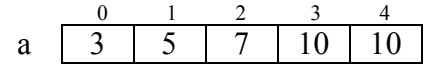
i = 0, t = 3
(0 > = 0) && (a[0] > 3) ⟶ TRUE



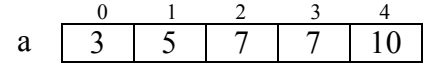
i = -1, t = 3
(-1 > = 0) && (a[-1] > 3) ⟶ FALSE



Pass four: j = 4, i = 3, t = a[4] = 2
(3 > = 0) && (a[3] > 2) ⟶ TRUE



i = 2, t = 2
(2 > = 0) && (a[2] > 2) ⟶ TRUE



i = 1, t = 2
(1 > = 0) && (a[1] > 2) ⟶ TRUE

	0	1	2	3	4
a	3	5	5	7	10

i = 0, t = 2
(0 > = 0) && (a[0] > 2) ⟶ TRUE

	0	1	2	3	4
a	3	3	5	7	10

i = -1, t = 2
(-1 > = 0) && (a[-1] > 2) ⟶ FALSE

	0	1	2	3	4
a	2	3	5	7	10

Bubble sort:

```
void main()
{
        int a[50], i, j, n, temp;
        scanf("%d",&n);

        for( i = 0; i < n; i ++)
                scanf("%d", & a[i]);

        for( i = 0; i < n-1; i ++)
                for( j = i+1; j < n; j ++)
                        if( a[i] > a[j]){
                                temp = a[ i ];
                                a[i] = a[ j ];
                                a[ j ] = temp;
                        }

        for( i = 0; i < n; i ++)
                printf("%d", a[i]);
}
```

⟵ Shorting logic

Example: Suppose that we have a list of 5 elements. We will use the bubble sort algorithm to sort the given list in ascending order.

	0	1	2	3	4
a	10	5	7	3	2

Pass one: i = 0, j = 1
(a[0] > a[1]) ⟶ TRUE

	0	1	2	3	4
a	5	10	7	3	2

j = 2
(a[0] > a[2]) ⟶ FALSE

	0	1	2	3	4
a	5	10	7	3	2

j = 3
(a[0] > a[3]) ⟶ TRUE

	0	1	2	3	4
a	3	10	7	5	2

j = 4
(a[0] > a[4]) ⟶ TRUE

	0	1	2	3	4
a	2	10	7	5	3

Pass two: i = 1, j = 2
(a[1] > a[2]) ⟶ TRUE

	0	1	2	3	4
a	2	7	10	5	3

j = 3
(a[1] > a[3]) ⟶ TRUE

	0	1	2	3	4
a	2	5	10	7	3

j = 4
(a[1] > a[4]) ⟶ TRUE

	0	1	2	3	4
a	2	3	10	7	5

Pass three: i = 2, j = 3
(a[2] > a[3]) ⟶ TRUE

	0	1	2	3	4
a	2	3	7	10	5

j = 4
(a[2] > a[4]) ⟶ TRUE

	0	1	2	3	4
a	2	3	5	10	7

Pass four: i = 3, j = 4
(a[3] > a[4]) ⟶ TRUE

	0	1	2	3	4
a	2	3	5	7	10