## Bubble Sort

|  |  |
| --- | --- |
| init | 1 4 3 9 5 8 7 |
| 1 | 1 4 3 5 8 7 **9** |
| 2 | 1 4 3 5 7 **8** 9 |
| 3 | 1 3 4 5 **7** 8 9 |
| 4 | 1 3 4 **5** 7 8 9 |
| 5 | 1 3 **4** 5 7 8 9 |
| 6 | 1 **3** 4 5 7 8 9 |

### Algorithm

Each time bubble the largest element to the right side.

For array A[n]:

The first round bubble A[0 .. n-1], result in the max element bubbled to A[n-1]

The second round bubble A[0 .. n-2], result in the max element bubbled to A[n-2]

…

The last (n-1) round bubble A[0 .. 1], result in the max element bubbled to A[1]

### Pseudocode

Bubble Sort A[n]

For i = n-1 to 1 (total rounds: n-1)

For j = 0 to i

If (A[j] < A[j + 1])

Swap(A[j], A[j+1])

## Insertion Sort

|  |  |
| --- | --- |
| init | 1 4 3 9 5 8 7 |
| 1 | 1 **4** 3 9 5 8 7 |
| 2 | 1 **3** 4 9 5 8 7 |
| 3 | 1 3 4 **9** 5 8 7 |
| 4 | 1 3 4 **5** 9 8 7 |
| 5 | 1 3 4 5 **8** 9 7 |
| 6 | 1 3 4 5 **7** 8 9 |

### Algorithm

Each time put A[n] into sorted sequence A[0 .. n-1]

The first round put A[1] into sorted sequence A[0]

The second round put A[2] into sorted sequence A[0..1]

..

The last round put A[n-1] into sorted sequence A[0..n-2]

### Pseudocode

Insertion Sort A[n]

For i = 1 to n – 1

Key = A[i]

While i > 0 && key < A[i - 1]

A[i - 1] = key

i--

A[i] = key

## Selection Sort

|  |  |
| --- | --- |
| init | 1 4 3 9 5 8 7 |
| 1 | **1** 4 3 9 5 8 7 |
| 2 | 1 **3** 4 9 5 8 7 |
| 3 | 1 3 **4** 9 5 8 7 |
| 4 | 1 3 4 **5** 9 8 7 |
| 5 | 1 3 4 5 **7** 9 8 |
| 6 | 1 3 4 5 7 **8** 9 |

### Algorithm

The first round mark index(0) as minimal, find index(i) that have smallest value in index(0..n-1), swap(0,i)

### Pseudocode

Selection Sort A[n]

for i = 0 to n-2

min = i

for j = i+1 to n-1

if (A[j] < A[min])

min = j

swap(A[i], A[min])