

Applied Discrete Mathematics



Applied Discrete Mathematics

Lecture 6





Table of Contents

- Concept of Algorithms.
- Searching Algorithms.
- **Sorting Algorithms.**





Sorting Algorithms.

Introduction (1/2)

Sorting is putting the elements into a list in which the elements are in increasing order.

For instance, Sorting the list **7, 2, 1, 4, 5, 9**

produces the list **1, 2, 4, 5, 7, 9**.

Sorting the list ***d, h, c, a, f***

produces the list ***a, c, d, f, h***.





Sorting Algorithms.

Introduction (2/2)

More than 100 sorting algorithms have been devised, and it is surprising how often new sorting algorithms are developed.

We will introduce **insertion** sort algorithms,

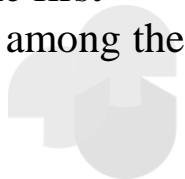




Sorting Algorithms.

The Insertion Sort

- The insertion sort is a simple sorting algorithm, but it is usually not the most efficient. To sort a list with n elements, the insertion sort **begins with the second element**. The insertion sort compares this second element.
- The insertion sort compares this second element with the first element and inserts it before the first element if it does not exceed the first element and after the first element if it exceeds the first element.
- At this point, the first two elements are in the correct order.
- **The third element is then compared with the first element**, and if it is larger than the first element, it is compared with the second element; it is inserted into the correct position among the first three elements.





Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

3	2	4	1	5
---	---	---	---	---



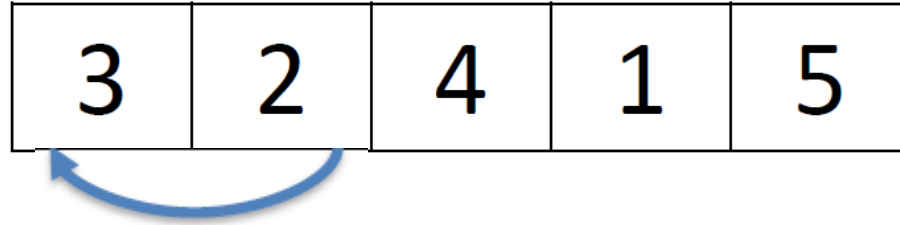


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

First step



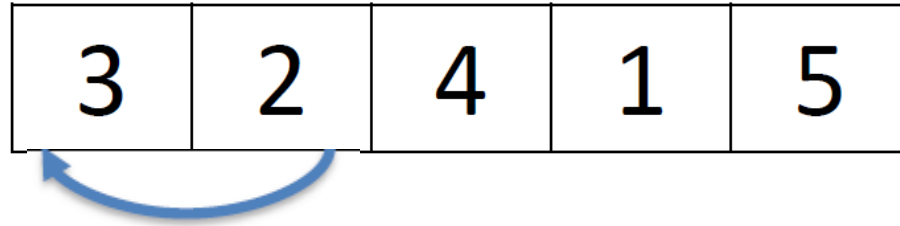


Sorting Algorithms.

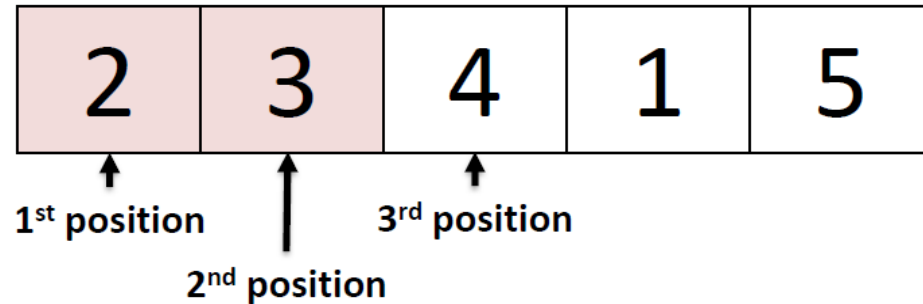
The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

First step



Because $2 < 3$,
it **inserts** 2 in the
first position.





Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Second step

2	3	4	1	5
---	---	---	---	---

sorted part



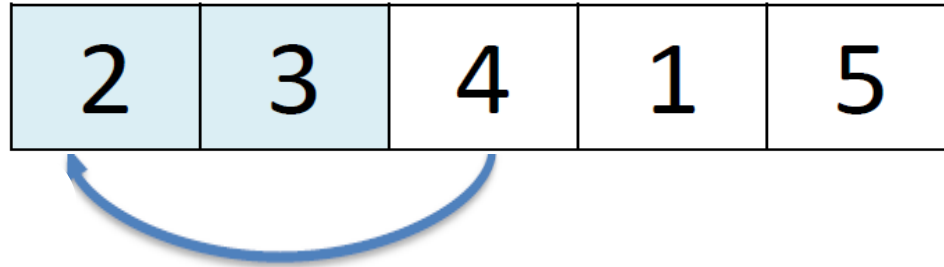


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Second step



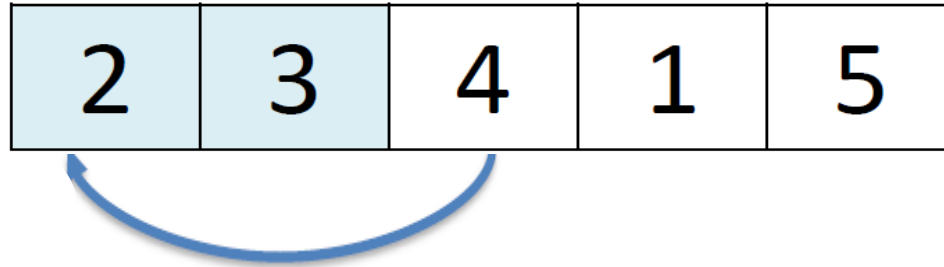


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Second step



Because $4 > 2$, then
compare with the
next element in the
sorted part



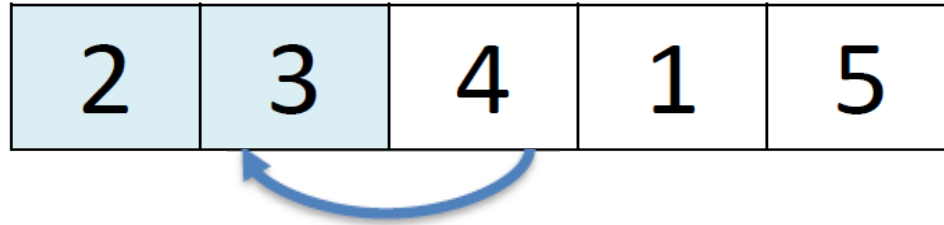


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Second step



Because $4 > 3$, then
compare with the
next element in the
sorted part



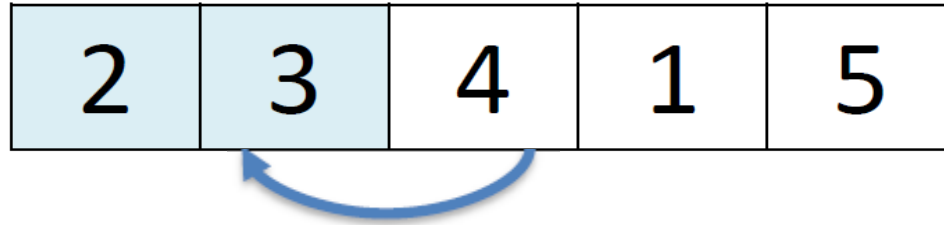


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Second step



Because $4 > 3$, then **compare** with the next element in the sorted part

Because there is no another element in the sorted part, then 4 **remains** in the 3rd location





Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Second step



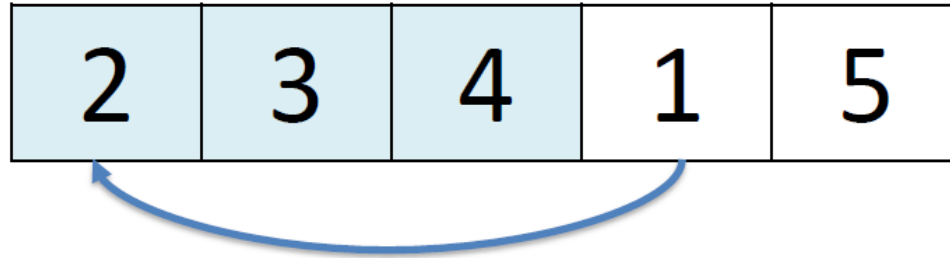


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Third step





Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Third step



Because $1 < 2$,
it **inserts** 1 in the
first position.



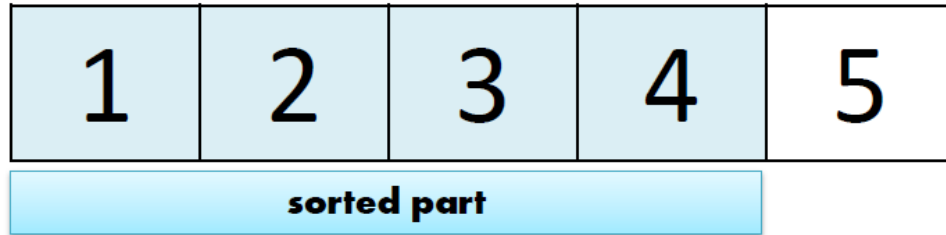


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Third step



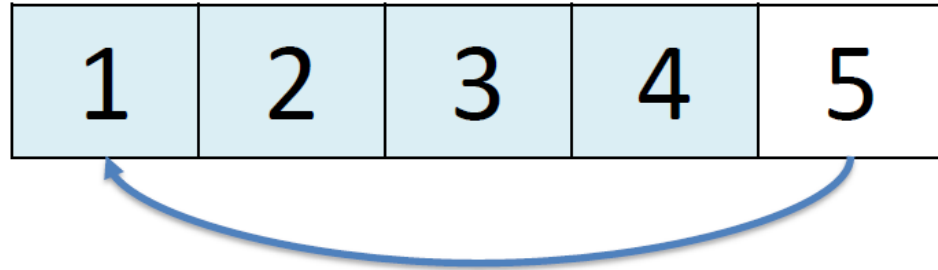


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Fourth step





Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Fourth step



Because $5 > 1$, then
compare with the
next element in the
sorted part



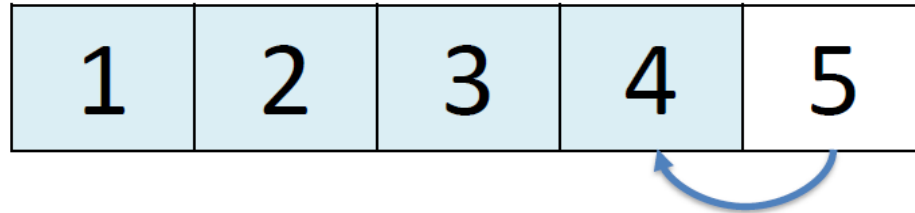


Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Fourth step



Because $5 > 4$, then **compare** with the next element in the sorted part

Because there is no another element in the sorted part, then 5 **remains** in the 5th location





Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Fourth step

1	2	3	4	5
---	---	---	---	---





Sorting Algorithms.

The Insertion Sort (Example 1)

Use the insertion sort to put 3, 2, 4, 1, 5 into increasing order.

Finally

1 2 3 4 5





Sorting Algorithms.

The Insertion Sort (Example 2)

Use the insertion sort to put

3, 44, 38, 5, 47, 15, 36, 26, 27, 2, 46, 4, 19, 50, 48

into increasing order.





Sorting Algorithms.

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

```
procedure insertion sort( $a_1, a_2, \dots, a_n$ : real numbers with  $n \geq 2$ )  
for  $j := 2$  to  $n$   
     $i := 1$   
    while  $a_j > a_i$   
         $i := i + 1$   
     $m := a_j$   
    for  $k := 0$  to  $j - i - 1$   
         $a_{j-k} := a_{j-k-1}$   
     $a_i := m$   
 $\{a_1, \dots, a_n$  is in increasing order}
```






Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

m	n	i	j	k
	5			



→ **procedure** *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order $\}$

1	5	2	4	3
---	---	---	---	---





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)
→ **for** $j := 2$ **to** n
 $i := 1$
 while $a_j > a_i$
 $i := i + 1$
 $m := a_j$
 for $k := 0$ **to** $j - i - 1$
 $a_{j-k} := a_{j-k-1}$
 $a_i := m$
{ a_1, \dots, a_n is in increasing order }

m	n	i	j	k
	5		2	

1	5	2	4	3
---	---	---	---	---





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

```
procedure insertion sort( $a_1, a_2, \dots, a_n$ : real numbers with  $n \geq 2$ )  
→ for  $j := 2$  to  $n$   $5$   
     $i := 1$   
    while  $a_j > a_i$   
         $i := i + 1$   
     $m := a_j$   
    for  $k := 0$  to  $j - i - 1$   
         $a_{j-k} := a_{j-k-1}$   
     $a_i := m$   
{  $a_1, \dots, a_n$  is in increasing order }
```

m	n	i	j	k
	5		2	

1	5	2	4	3
---	---	---	---	---





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$\rightarrow i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}

m	n	i	j	k
	5	1	2	



1	5	2	4	3
---	---	---	---	---





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

 ➔ **while** $a_j > a_i$
 $i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}

m	n	i	j	k
	5	1	2	

1	2	3	4	5
1	5	2	4	3
a_i	a_j			





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$ ✓
 $i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}

m	n	i	j	k
	5	1	2	

1	2	3	4	5
1	5	2	4	3
a_i	a_j			





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$\Rightarrow i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
	5	2	2	

1	2	3	4	5
1	5	2	4	3





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

 ➔ **while** $a_j > a_i$

$i := i + 1$

$m := a_j$

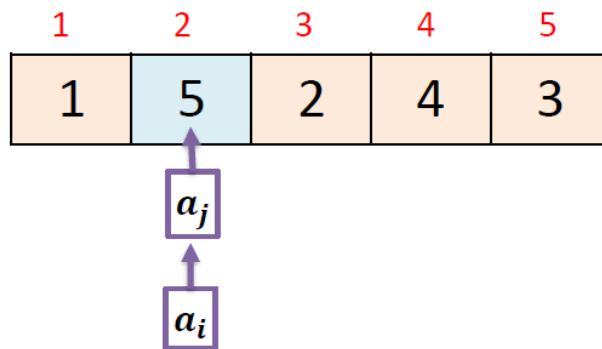
for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order }

m	n	i	j	k
	5	2	2	





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

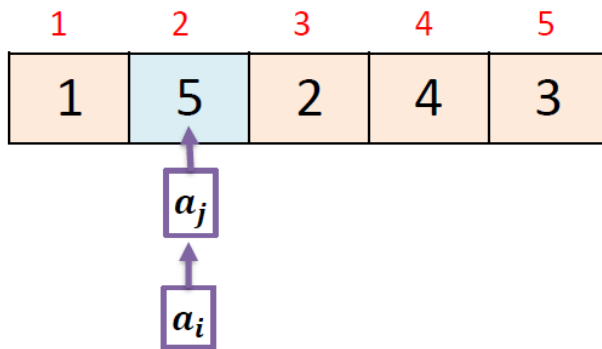
for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order }

m	n	i	j	k
	5	2	2	





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

 ➔ $m := a_j$

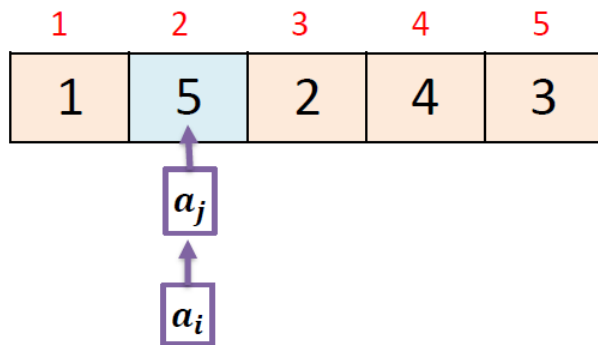
for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
5	5	2	2	





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$


 ➡ **for** $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
5	5	2	2	0



1	2	3	4	5
1	5	2	4	3





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

m	n	i	j	k
5	5	2	2	0

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

-1

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

1	2	3	4	5
1	5	2	4	3





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

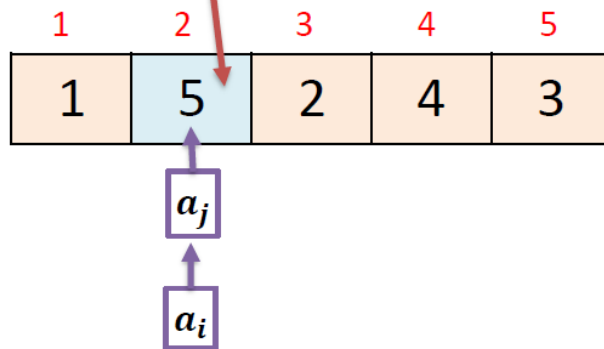
for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$\Rightarrow a_i := m$

$\{a_1, \dots, a_n$ is in increasing order $\}$

m	n	i	j	k
5	5	2	2	0






Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

m	n	i	j	k
5	5	2	3	0



procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)
→ **for** $j := 2$ **to** n
 $i := 1$
 while $a_j > a_i$
 $i := i + 1$
 $m := a_j$
 for $k := 0$ **to** $j - i - 1$
 $a_{j-k} := a_{j-k-1}$
 $a_i := m$
 $\{a_1, \dots, a_n$ is in increasing order}

1	2	3	4	5
1	5	2	4	3





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$\rightarrow i := 1$

while $a_j > a_i$
 $i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
5	5	1	3	0

1	2	3	4	5
1	5	2	4	3





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

 ➔ **while** $a_j > a_i$
 $i := i + 1$

$m := a_j$

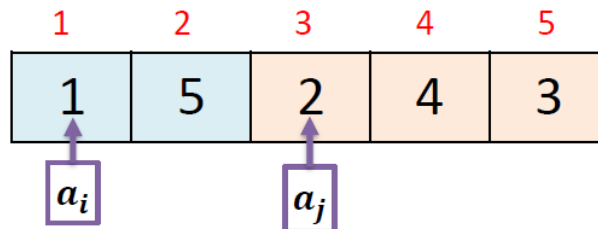
for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order }

m	n	i	j	k
5	5	1	3	0





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

 ➔ **while** $a_j \checkmark a_i$
 $i := i + 1$

$m := a_j$

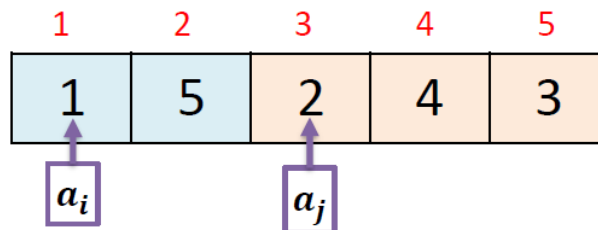
for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order }

m	n	i	j	k
5	5	1	3	0





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j \checkmark a_i$

$\rightarrow i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order $\}$

m	n	i	j	k
5	5	2	3	0

1	2	3	4	5
1	5	2	4	3

a_i a_j





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

m	n	i	j	k
5	5	2	3	0

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

 ➔ **while** $a_j \times a_i$
 $i := i + 1$

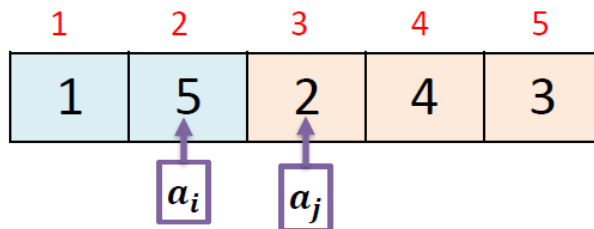
$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)
for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$
 $i := i + 1$

 ➔ $m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}

m	n	i	j	k
2	5	2	3	0

1	2	3	4	5
1	5	2	4	3

a_i a_j





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

 ➡ **for** $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
2	5	2	3	0

1	2	3	4	5
1	5	2	4	3

a_i a_j





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

m	n	i	j	k
2	5	2	3	0

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

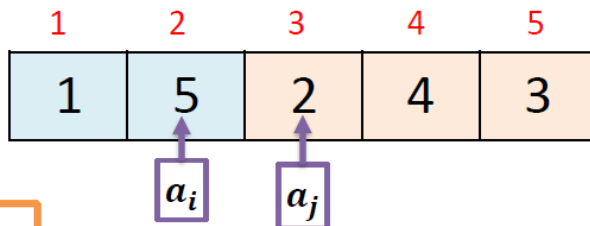
$m := a_j$

 ➔ **for** $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}



0





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

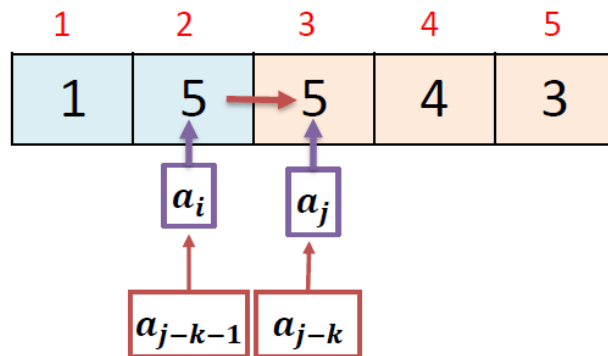
for $k := 0$ **to** $j - i - 1$

$\Rightarrow a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order $\}$

m	n	i	j	k
2	5	2	3	0





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

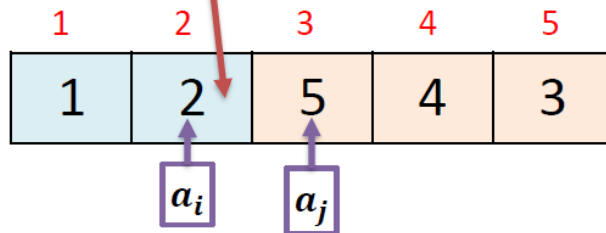
for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
2	5	2	3	0





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

 ➔ **while** $a_j > a_i$
 $i := i + 1$

$m := a_j$

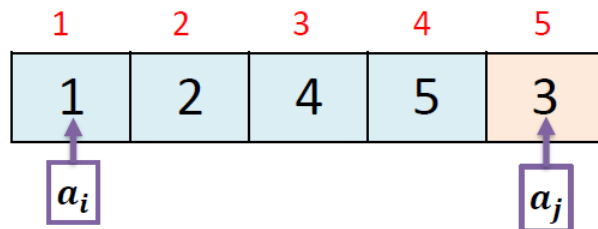
for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
4	5	1	5	0





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

 ➔ **while** $a_j > a_i$

$i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}

m	n	i	j	k
4	5	3	5	0

Continue

1	2	4	5	3
1	2	4	5	3
		a_i		a_j





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$
 $i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}

m	n	i	j	k
4	5	3	5	0

Continue

1	2	4	5	3
		a_i		a_j





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
3	5	3	5	0

1	2	4	5	3
		a_i		a_j



Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

 ➔ **for** $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
3	5	3	5	0

1	2	3	4	5
1	2	4	5	3

a_i points to the value 4 in the 3rd position.
 a_j points to the value 3 in the 5th position.





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

 ➔ **for** $k := 0$ **to** $j - i - 1$

1

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
3	5	3	5	0

1	2	3	4	5
1	2	4	5	3

a_i points to the value 4 in the 3rd position.
 a_j points to the value 3 in the 5th position.





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

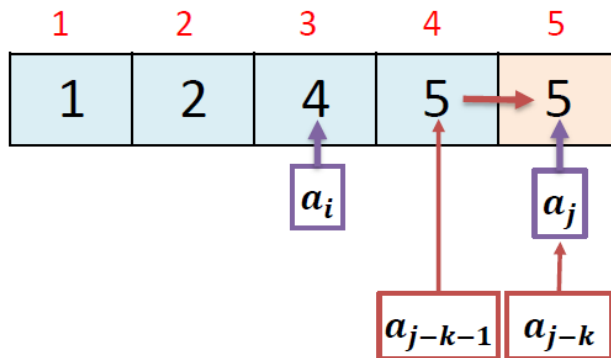
for $k := 0$ **to** $j - i - 1$

$\Rightarrow a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}

m	n	i	j	k
3	5	3	5	0





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

 ➔ **for** $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

{ a_1, \dots, a_n is in increasing order}

m	n	i	j	k
3	5	3	5	1

1	2	3	4	5
1	2	4	5	5

a_i points to the 4 in the 3rd position.
 a_j points to the 5 in the 5th position.





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

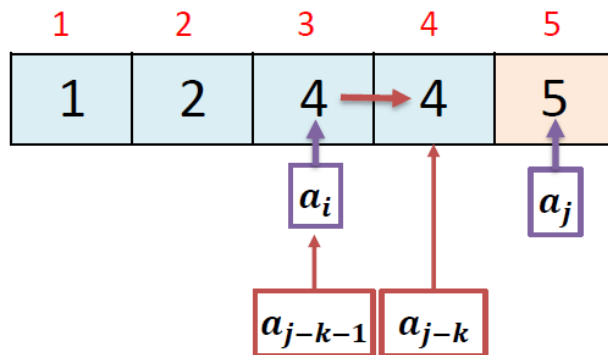
for $k := 0$ **to** $j - i - 1$

$\Rightarrow a_{j-k} := a_{j-k-1}$

$a_i := m$

$\{a_1, \dots, a_n$ is in increasing order $\}$

m	n	i	j	k
3	5	3	5	1





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$\Rightarrow a_i := m$

$\{a_1, \dots, a_n$ is in increasing order}

m	n	i	j	k
3	5	3	5	1

1	2	3	4	5
1	2	3	4	5

a_i a_j





Sorting Algorithms.

Example 1

The Insertion Sort (Algorithm)

m	n	i	j	k
3	5	3	5	1

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*(a_1, a_2, \dots, a_n : real numbers with $n \geq 2$)

for $j := 2$ **to** n

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

→ { a_1, \dots, a_n is in increasing order}

1	2	3	4	5
1	2	3	4	5

