Insertion Sort

Parosh Aziz Abdulla

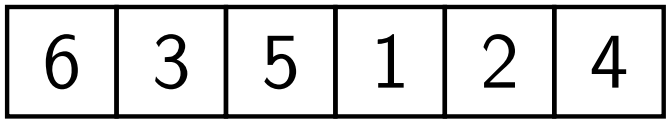
Uppsala University

July 24, 2012

Insertion Sort

INSERTION-SORT(A)

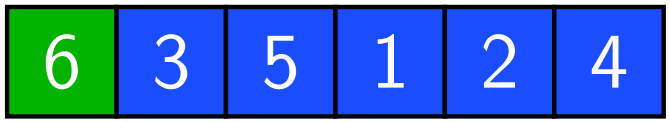
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

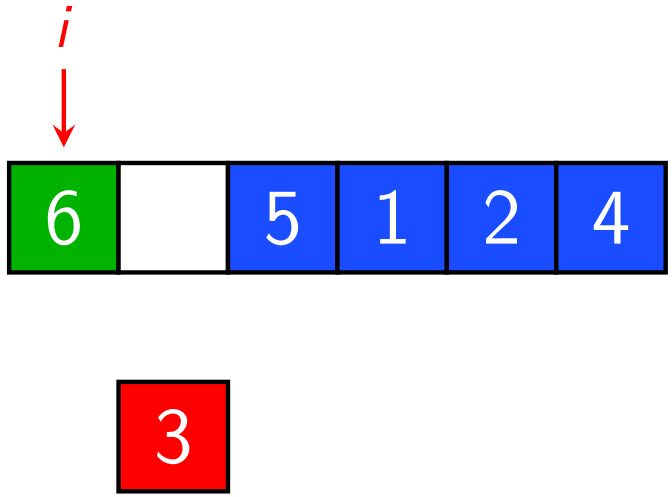


$j = 2$

Insertion Sort

INSERTION-SORT(A)

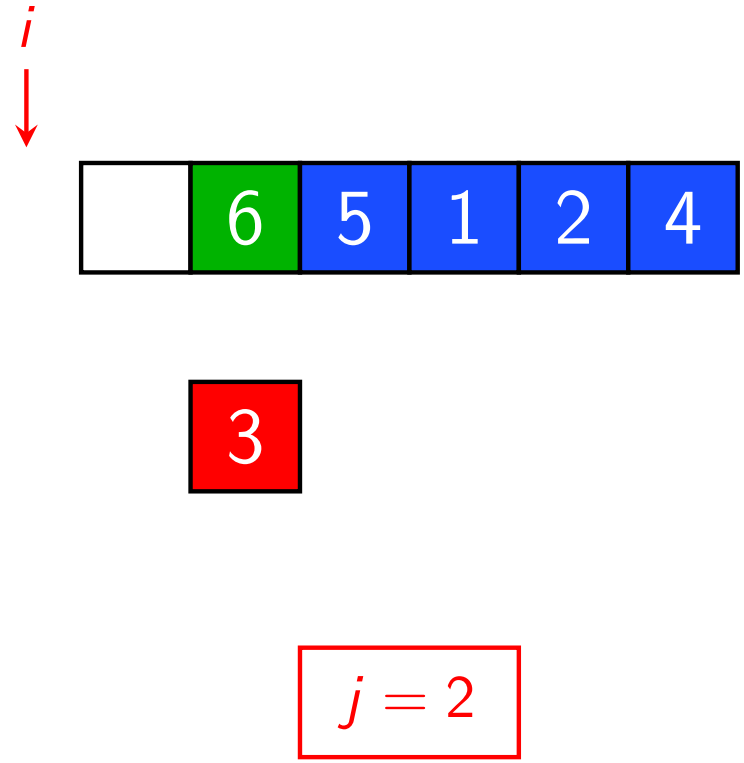
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

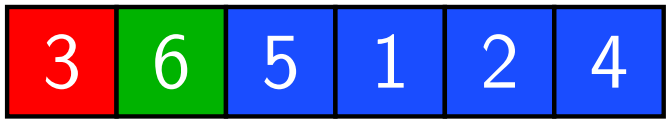
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

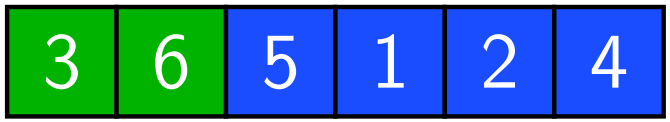


$j = 2$

Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

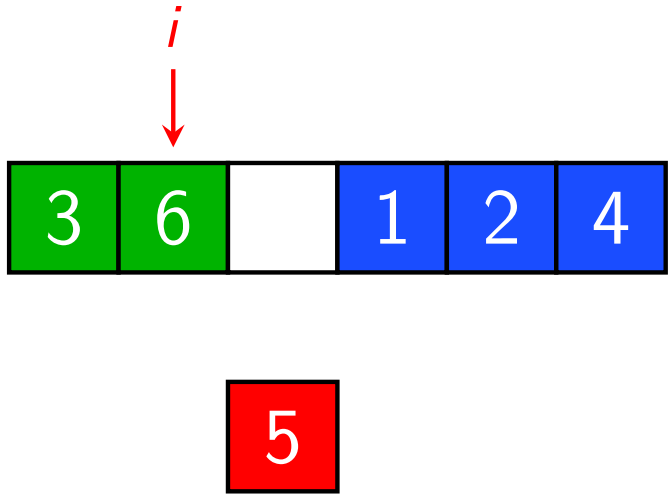


$j = 3$

Insertion Sort

INSERTION-SORT(A)

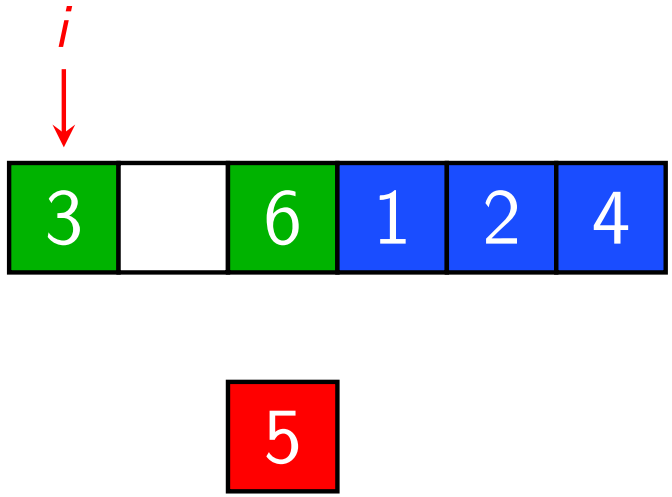
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

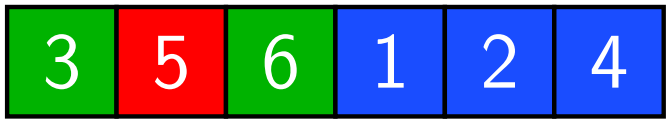
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

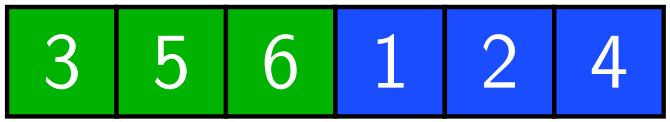


$j = 3$

Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

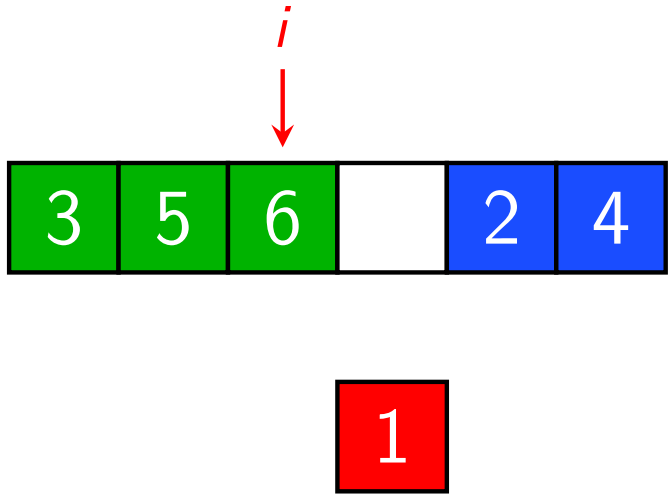


$j = 4$

Insertion Sort

INSERTION-SORT(A)

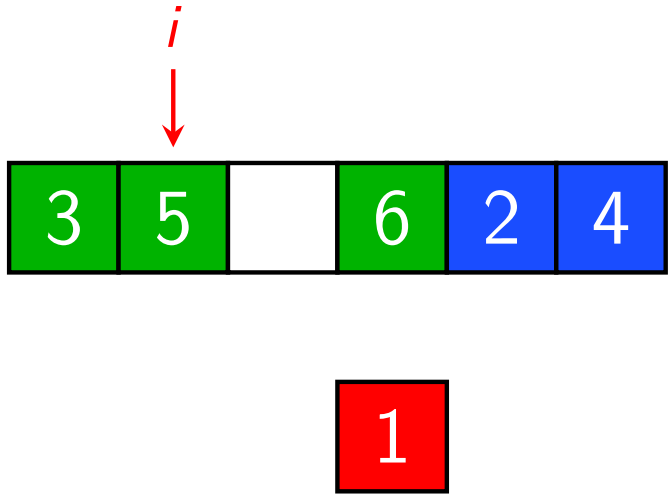
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

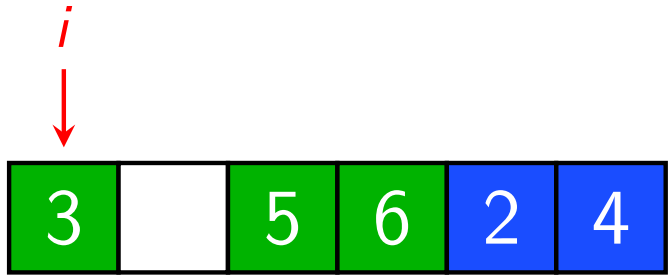
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

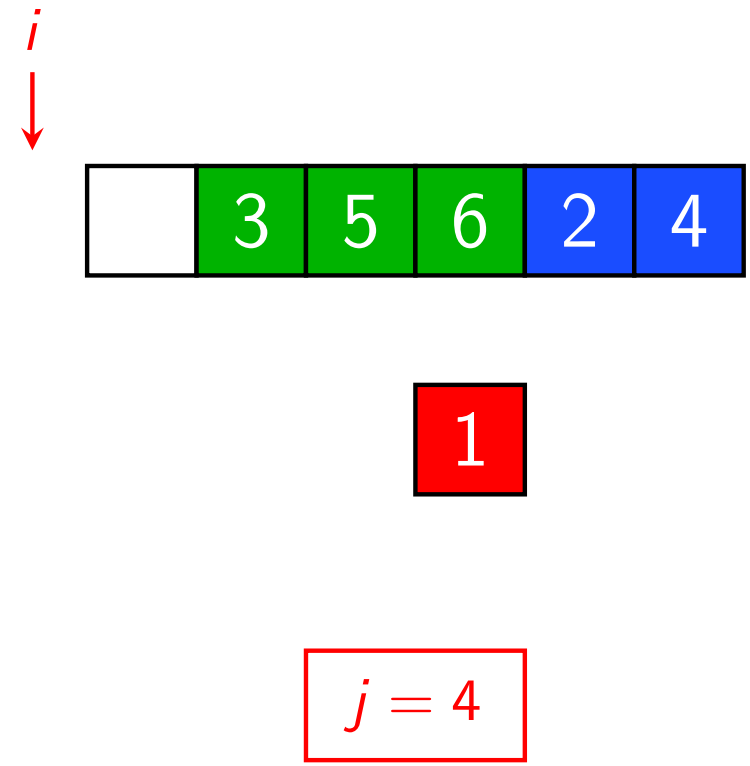


$j = 4$

Insertion Sort

INSERTION-SORT(A)

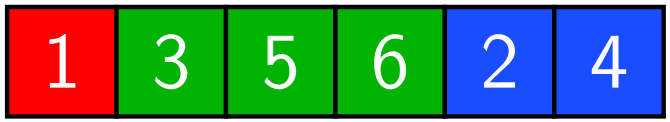
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

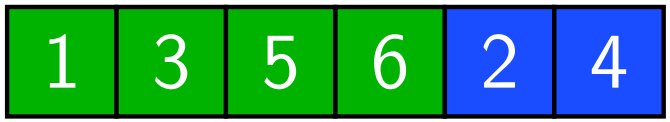


$j = 4$

Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

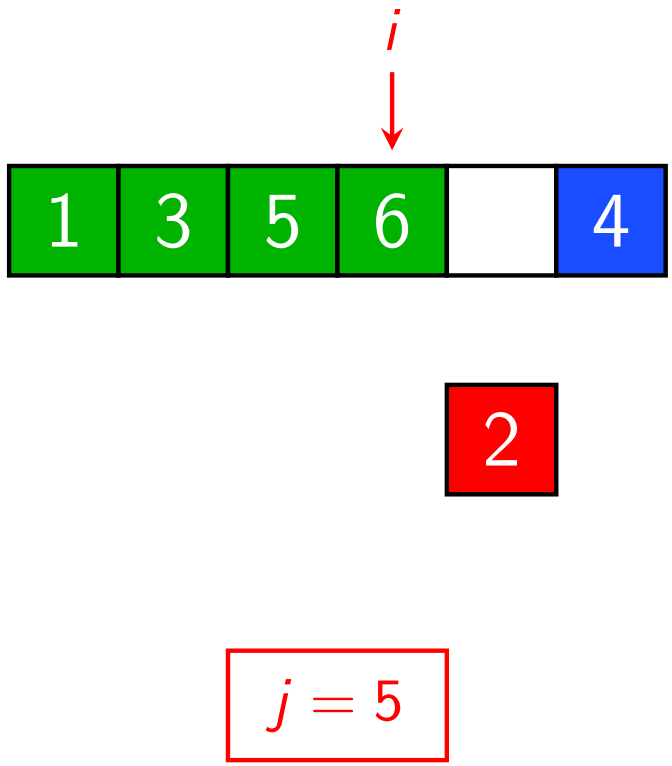


$j = 5$

Insertion Sort

INSERTION-SORT(A)

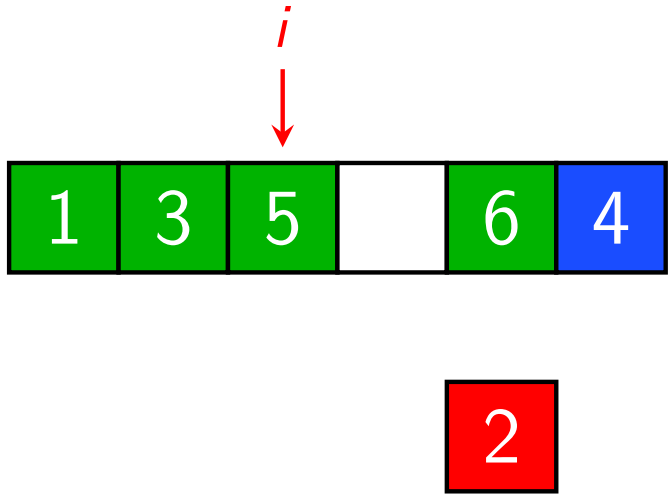
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

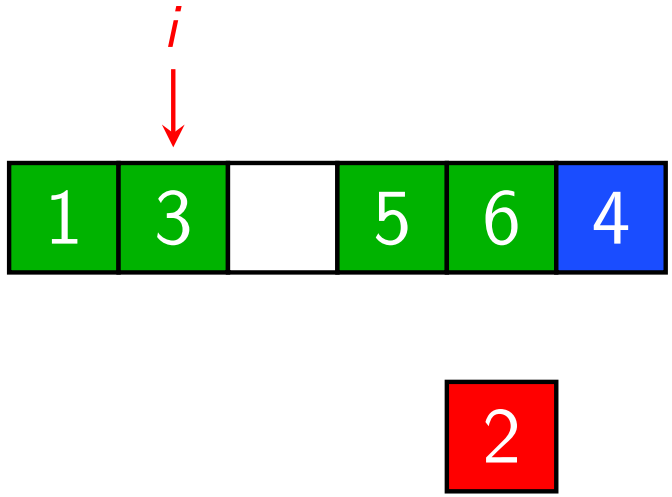
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

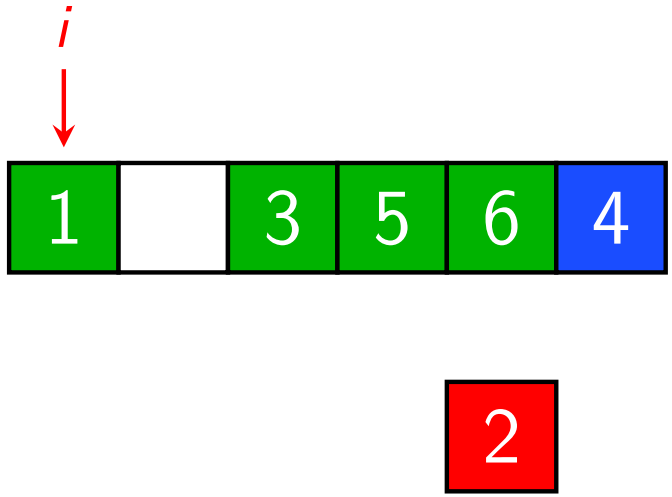
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

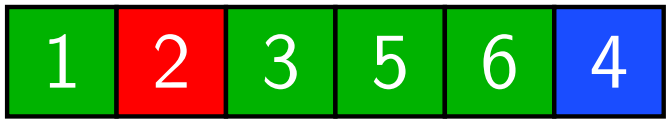
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

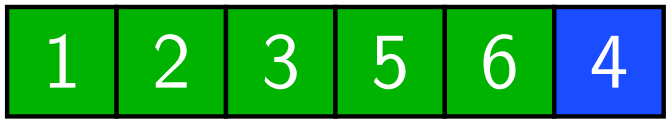


$j = 5$

Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

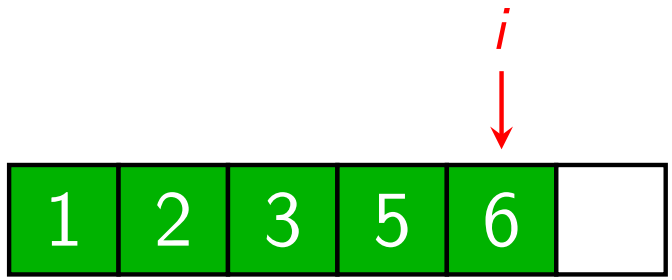


$j = 6$

Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

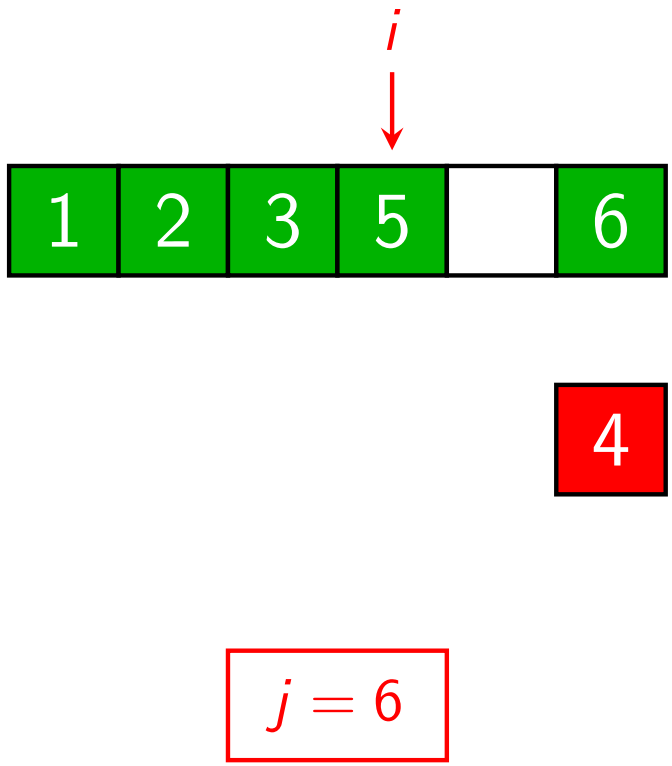


$j = 6$

Insertion Sort

INSERTION-SORT(A)

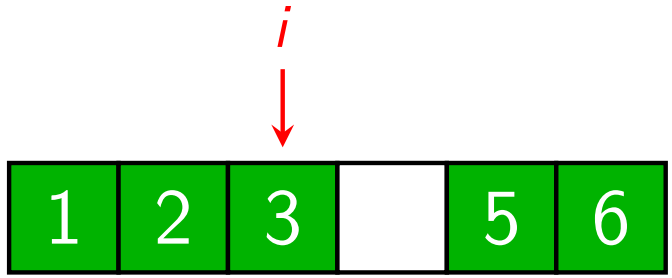
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

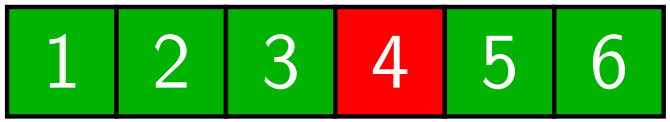


$j = 6$

Insertion Sort

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



$j = 6$

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$		
2 **do** $key \leftarrow A[j]$		
3 $\triangleright$ Insert $A[j]$ into $A[1..j-1]$.		
4 $i \leftarrow j - 1$		
5 **while** $i > 0$ and $A[i] > key$		
6 **do** $A[i+1] \leftarrow A[i]$		
7 $i \leftarrow i - 1$		
8 $A[i+1] \leftarrow key$		

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	
2 **do** $key \leftarrow A[j]$	c_2	
3 ▷ Insert $A[j]$ into $A[1..j-1]$.	0	
4 $i \leftarrow j-1$	c_4	
5 **while** $i > 0$ and $A[i] > key$	c_5	
6 **do** $A[i+1] \leftarrow A[i]$	c_6	
7 $i \leftarrow i-1$	c_7	
8 $A[i+1] \leftarrow key$	c_8	

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	n
2 **do** $key \leftarrow A[j]$	c_2	
3 ▷ Insert $A[j]$ into $A[1..j-1]$.	0	
4 $i \leftarrow j-1$	c_4	
5 **while** $i > 0$ and $A[i] > key$	c_5	
6 **do** $A[i+1] \leftarrow A[i]$	c_6	
7 $i \leftarrow i-1$	c_7	
8 $A[i+1] \leftarrow key$	c_8	

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	n
2 **do** $key \leftarrow A[j]$	c_2	$n-1$
3 ▷ Insert $A[j]$ into $A[1..j-1]$.	0	
4 $i \leftarrow j-1$	c_4	
5 **while** $i > 0$ and $A[i] > key$	c_5	
6 **do** $A[i+1] \leftarrow A[i]$	c_6	
7 $i \leftarrow i-1$	c_7	
8 $A[i+1] \leftarrow key$	c_8	

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	n
2 **do** $key \leftarrow A[j]$	c_2	$n-1$
3 $\triangleright$ Insert $A[j]$ into $A[1..j-1]$.	0	$n-1$
4 $i \leftarrow j-1$	c_4	
5 **while** $i > 0$ and $A[i] > key$	c_5	
6 **do** $A[i+1] \leftarrow A[i]$	c_6	
7 $i \leftarrow i-1$	c_7	
8 $A[i+1] \leftarrow key$	c_8	

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	n
2 **do** $key \leftarrow A[j]$	c_2	$n-1$
3 $\triangleright$ Insert $A[j]$ into $A[1..j-1]$.	0	$n-1$
4 $i \leftarrow j-1$	c_4	$n-1$
5 **while** $i > 0$ and $A[i] > key$	c_5	
6 **do** $A[i+1] \leftarrow A[i]$	c_6	
7 $i \leftarrow i-1$	c_7	
8 $A[i+1] \leftarrow key$	c_8	

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	n
2 **do** $key \leftarrow A[j]$	c_2	$n-1$
3 $\triangleright$ Insert $A[j]$ into $A[1..j-1]$.	0	$n-1$
4 $i \leftarrow j-1$	c_4	$n-1$
5 **while** $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6 **do** $A[i+1] \leftarrow A[i]$	c_6	
7 $i \leftarrow i-1$	c_7	
8 $A[i+1] \leftarrow key$	c_8	

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	n
2 **do** $key \leftarrow A[j]$	c_2	$n-1$
3 $\triangleright$ Insert $A[j]$ into $A[1..j-1]$.	0	$n-1$
4 $i \leftarrow j-1$	c_4	$n-1$
5 **while** $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6 **do** $A[i+1] \leftarrow A[i]$	c_6	$\sum_{j=2}^{n} (t_j - 1)$
7 $i \leftarrow i-1$	c_7	
8 $A[i+1] \leftarrow key$	c_8	

Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	n
2 **do** $key \leftarrow A[j]$	c_2	$n-1$
3 $\triangleright$ Insert $A[j]$ into $A[1..j-1]$.	0	$n-1$
4 $i \leftarrow j-1$	c_4	$n-1$
5 **while** $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6 **do** $A[i+1] \leftarrow A[i]$	c_6	$\sum_{j=2}^{n} (t_j - 1)$
7 $i \leftarrow i-1$	c_7	$\sum_{j=2}^{n} (t_j - 1)$
8 $A[i+1] \leftarrow key$	c_8	

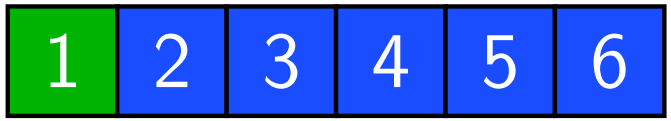
Insertion Sort - Cost Analysis

INSERTION-SORT(A)	cost	times
1 **for** $j \leftarrow 2$ **to** $length[A]$	c_1	n
2 **do** $key \leftarrow A[j]$	c_2	$n-1$
3 $\triangleright$ Insert $A[j]$ into $A[1..j-1]$.	0	$n-1$
4 $i \leftarrow j-1$	c_4	$n-1$
5 **while** $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6 **do** $A[i+1] \leftarrow A[i]$	c_6	$\sum_{j=2}^{n} (t_j - 1)$
7 $i \leftarrow i-1$	c_7	$\sum_{j=2}^{n} (t_j - 1)$
8 $A[i+1] \leftarrow key$	c_8	$n-1$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

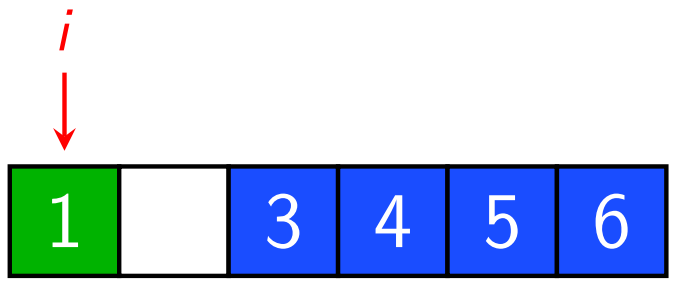


$j = 2$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



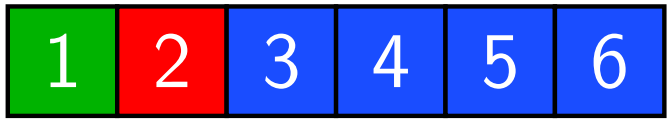
2

$j = 2$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



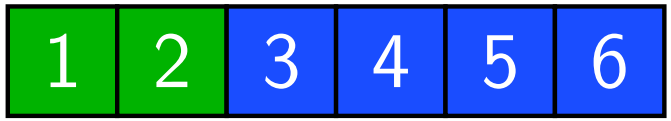
$j = 2$

$t_j = 1$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

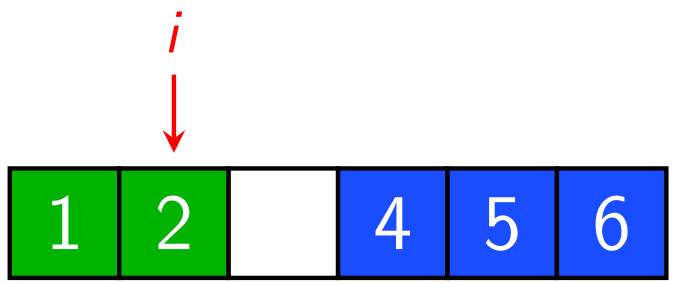


$j = 3$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



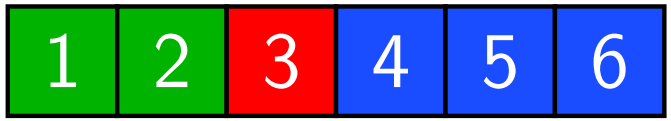
3

$j = 3$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



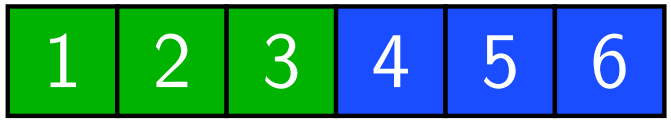
$j = 3$

$t_j = 1$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

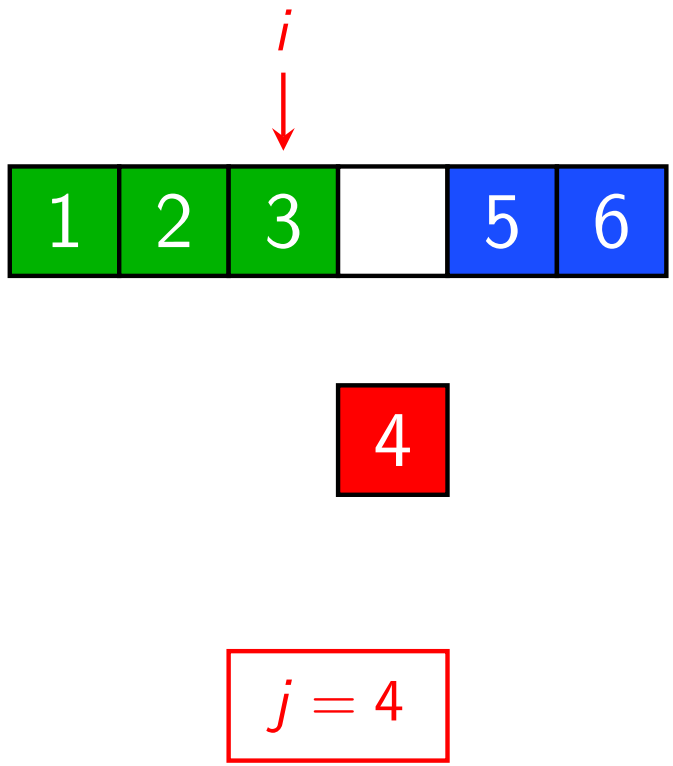


$j = 4$

Insertion Sort - Best Case

INSERTION-SORT(A)

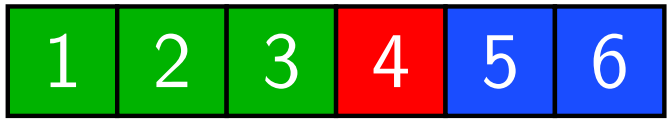
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



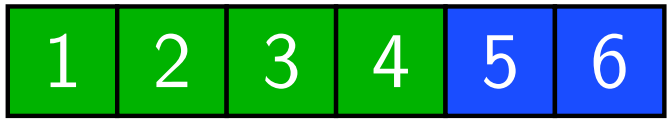
$j = 4$

$t_j = 1$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

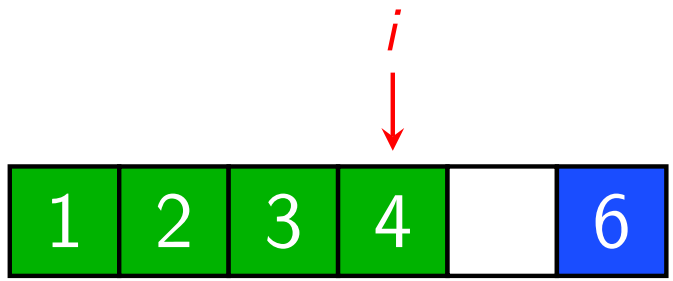


$j = 5$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

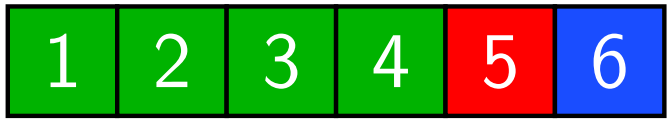


$j = 5$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



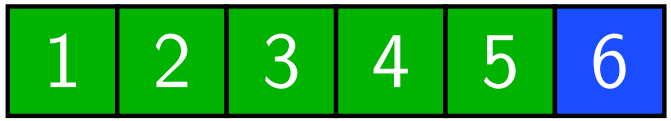
$j = 5$

$t_j = 1$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

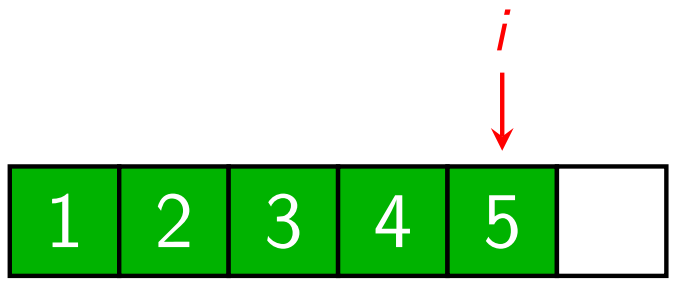


$j = 6$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



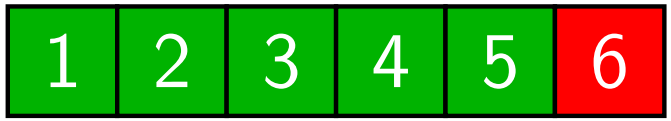
6

$j = 6$

Insertion Sort - Best Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



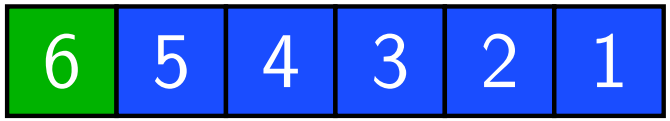
$j = 6$

$t_j = 1$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

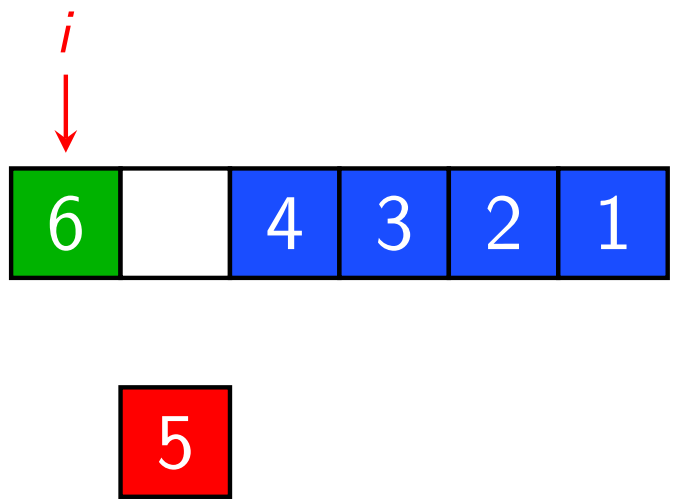


$j = 2$

Insertion Sort - Worst Case

INSERTION-SORT(A)

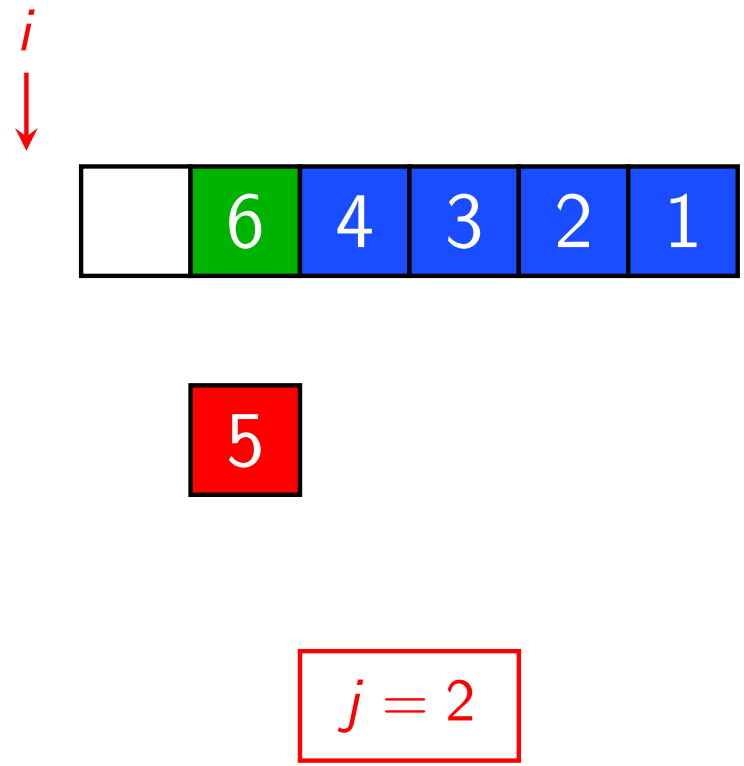
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Worst Case

INSERTION-SORT(A)

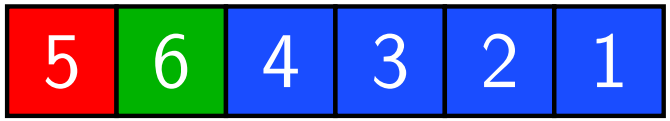
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



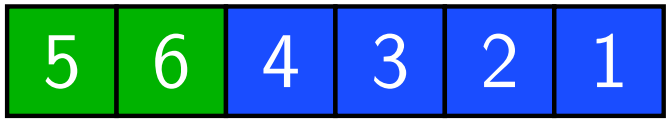
$j = 2$

$t_j = 2$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

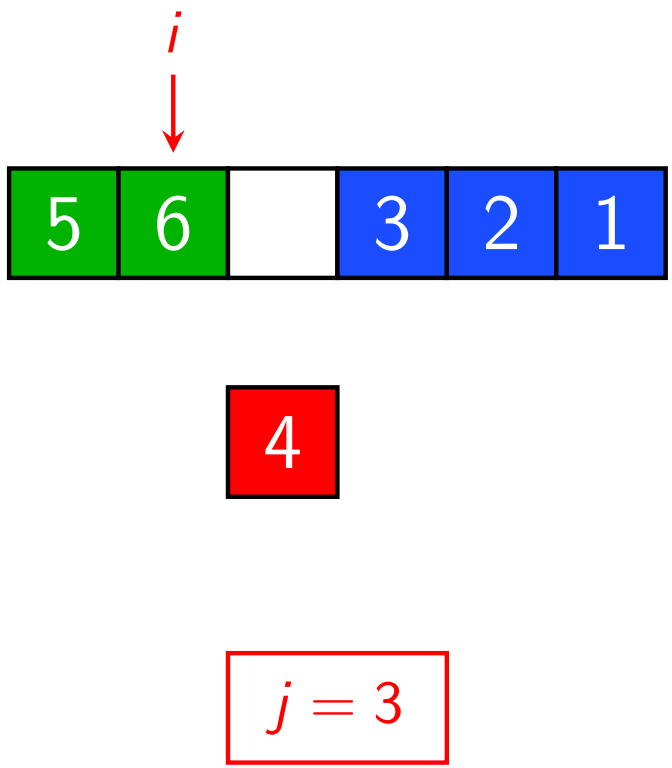


$j = 3$

Insertion Sort - Worst Case

INSERTION-SORT(A)

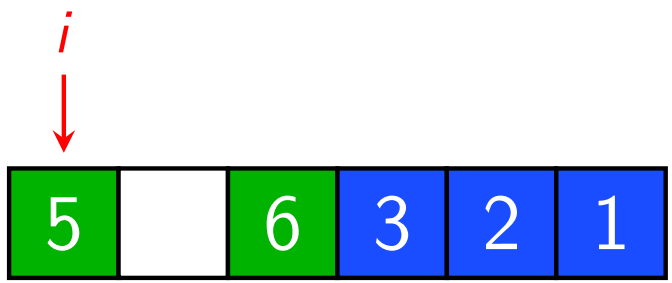
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



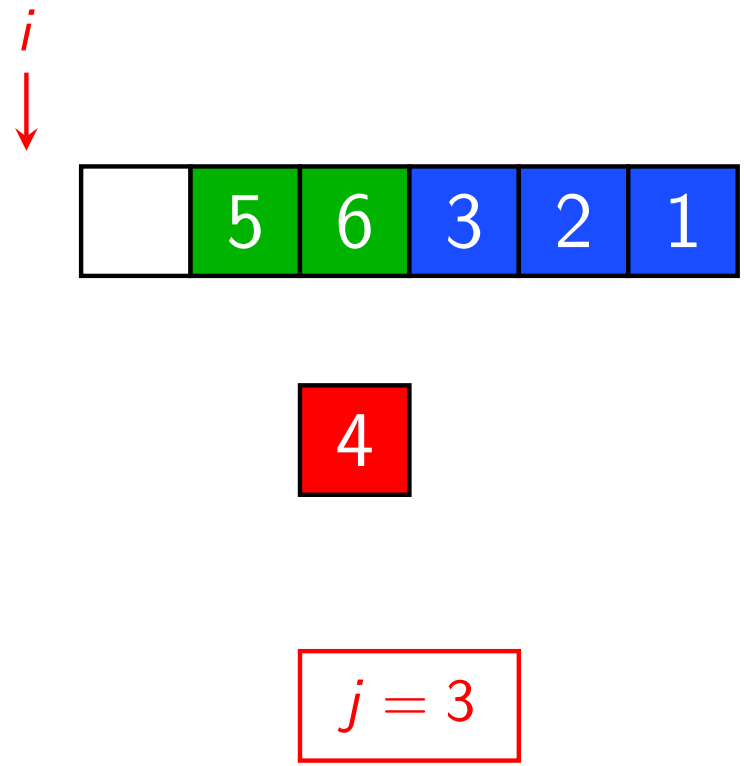
4

$j = 3$

Insertion Sort - Worst Case

INSERTION-SORT(A)

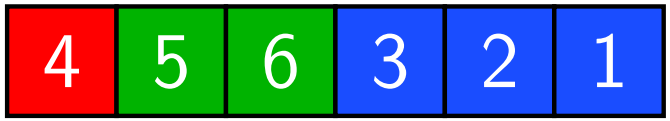
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



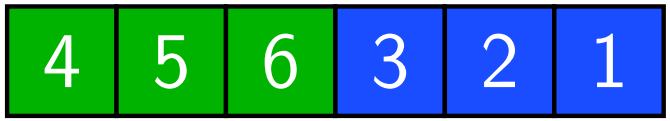
$j = 3$

$t_j = 3$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

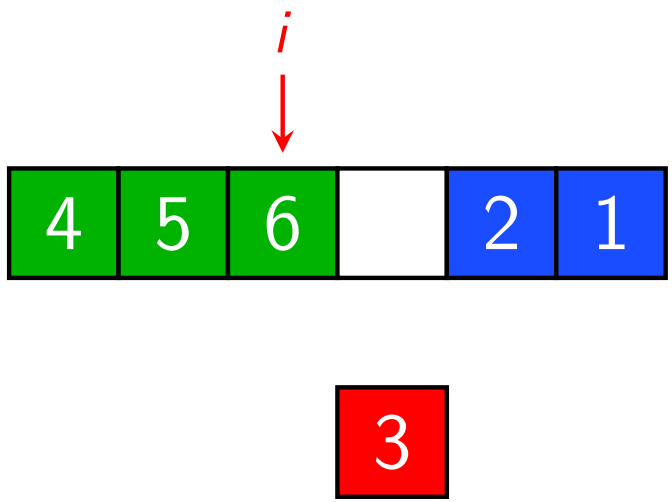


$j = 4$

Insertion Sort - Worst Case

INSERTION-SORT(A)

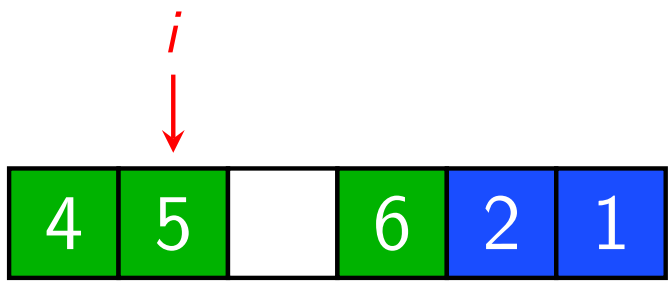
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



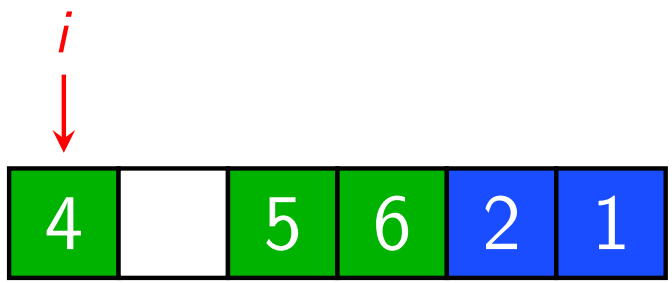
3

$j = 4$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



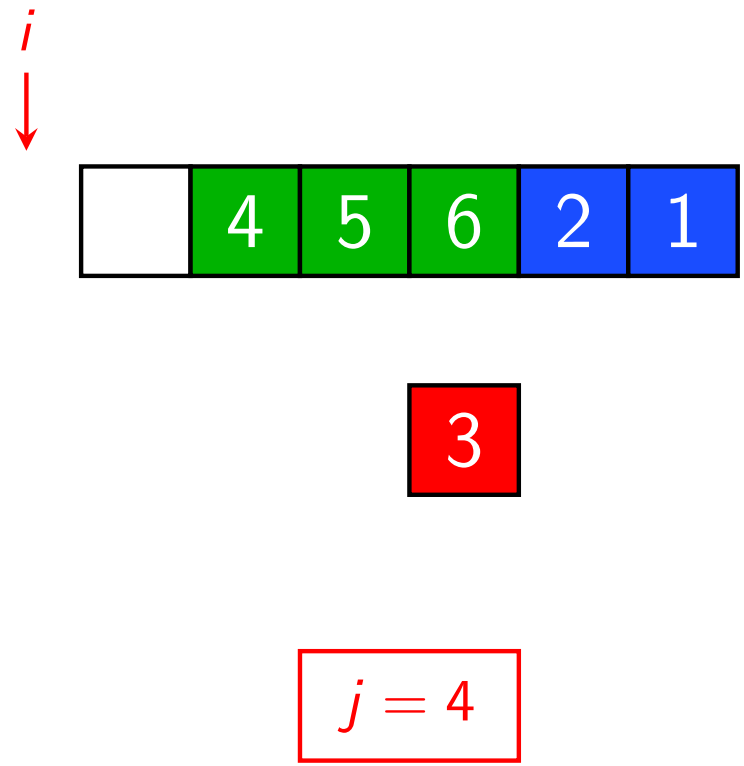
3

$j = 4$

Insertion Sort - Worst Case

INSERTION-SORT(A)

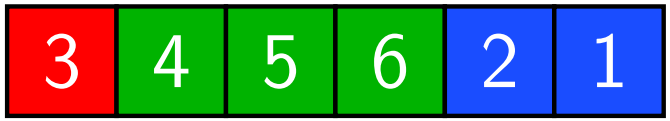
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



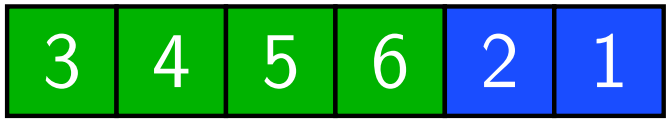
$j = 4$

$t_j = 4$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

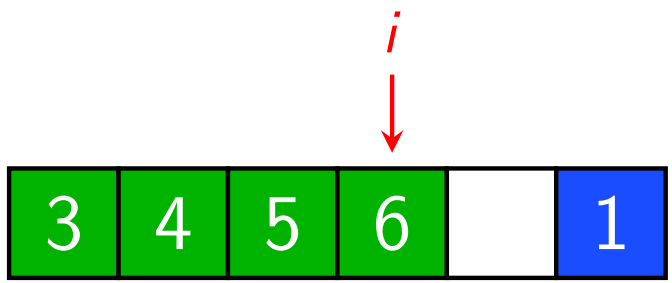


$j = 5$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

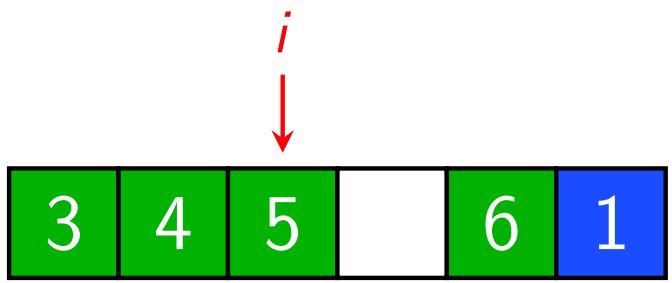


$j = 5$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

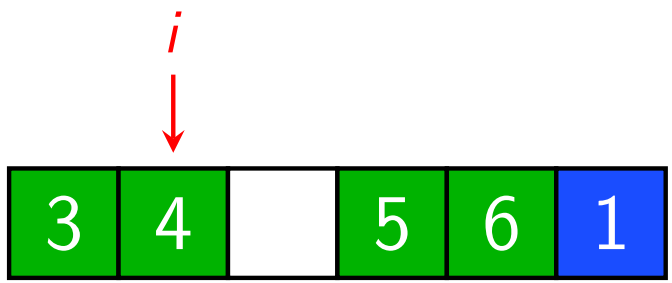


$j = 5$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



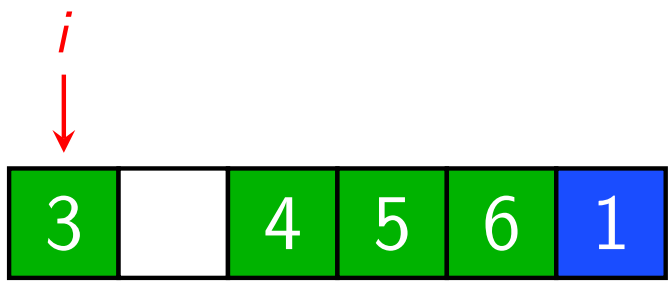
2

$j = 5$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



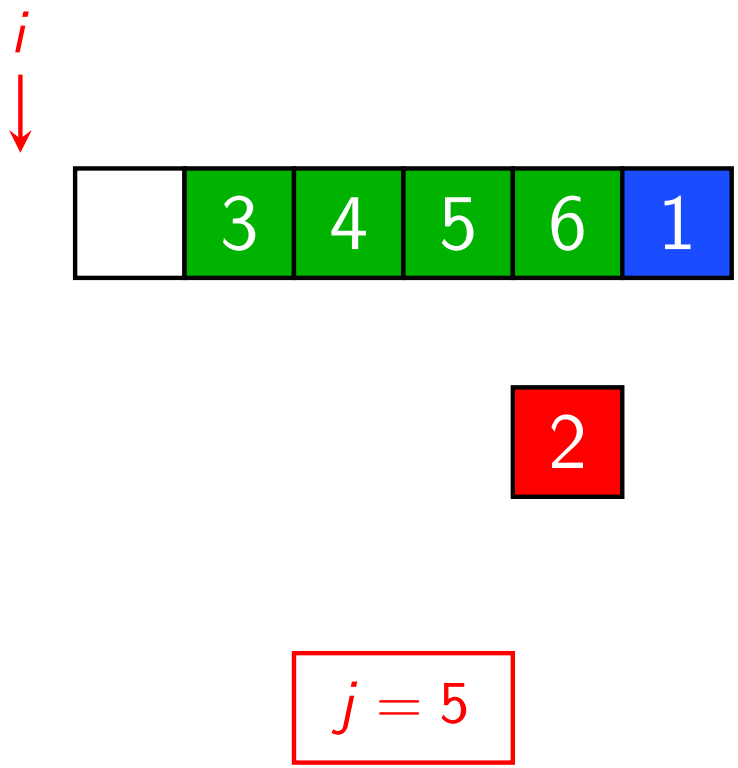
2

$j = 5$

Insertion Sort - Worst Case

INSERTION-SORT(A)

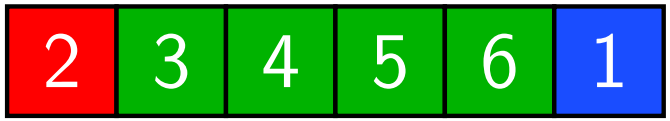
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



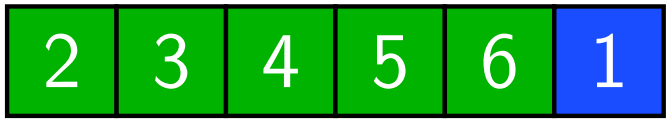
$j = 5$

$t_j = 5$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

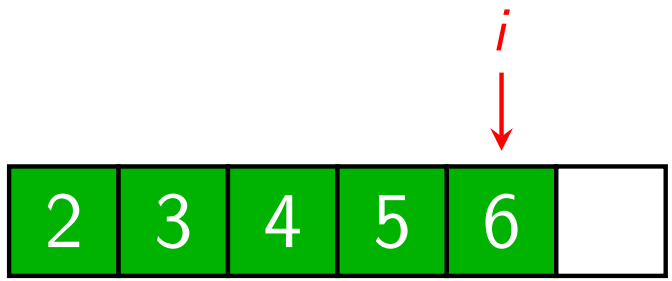


$j = 6$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

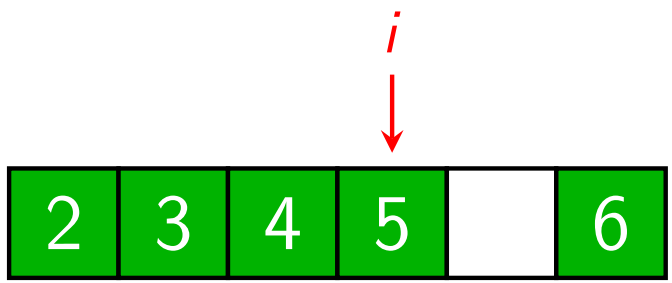


$j = 6$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

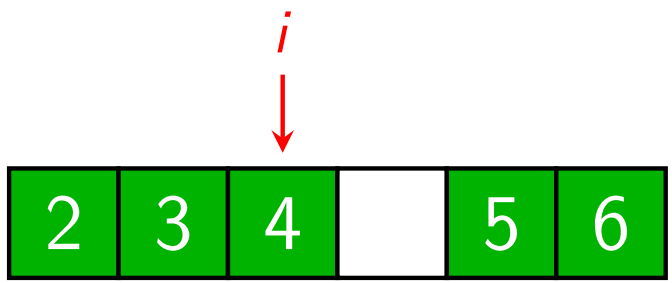


$j = 6$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

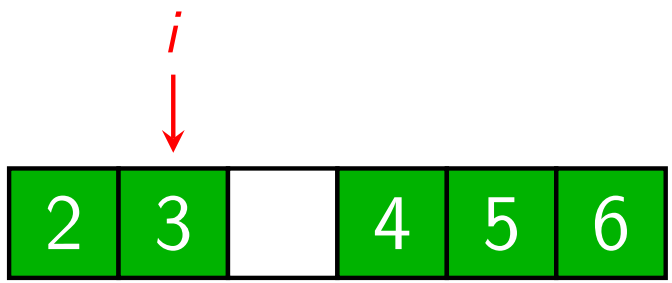


$j = 6$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



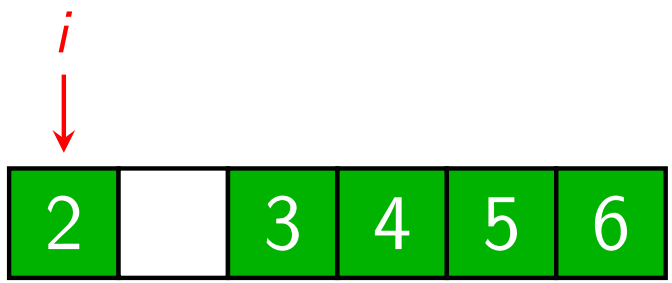
1

$j = 6$

Insertion Sort - Worst Case

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

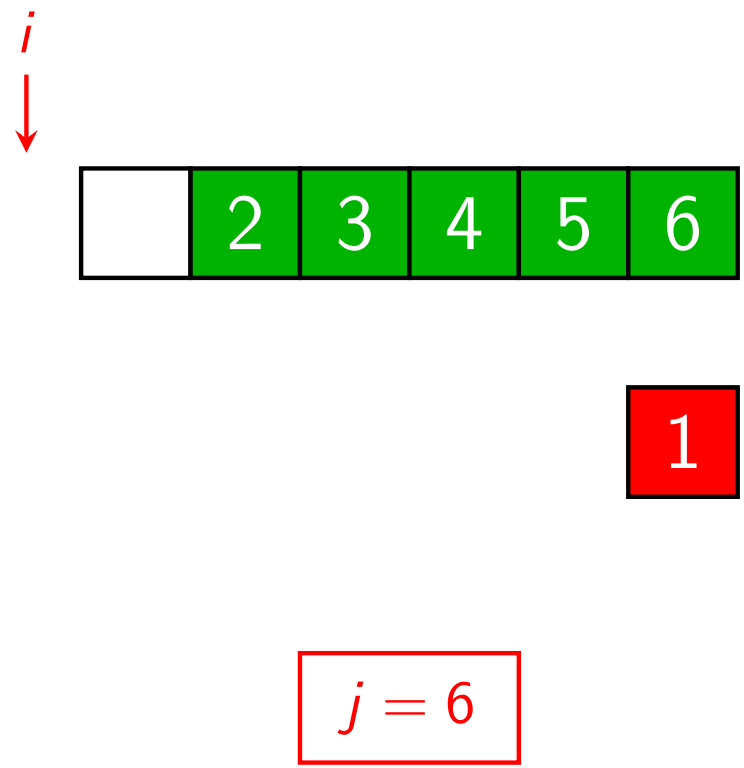


$j = 6$

Insertion Sort - Worst Case

INSERTION-SORT(A)

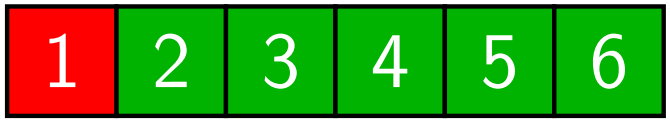
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



Insertion Sort - Worst Case

INSERTION-SORT(A)

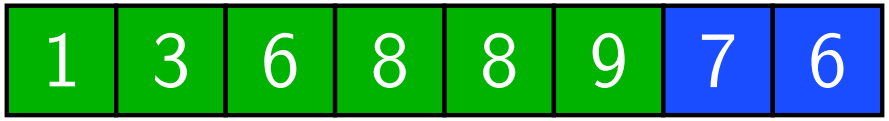
```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```



$j = 6$

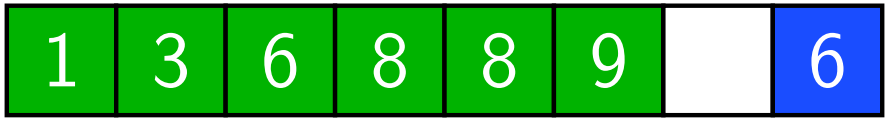
$t_j = 6$

Insertion Sort - Average Case



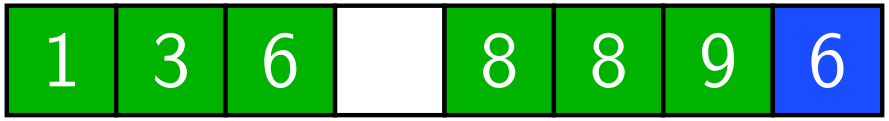
$j = 7$

Insertion Sort - Average Case



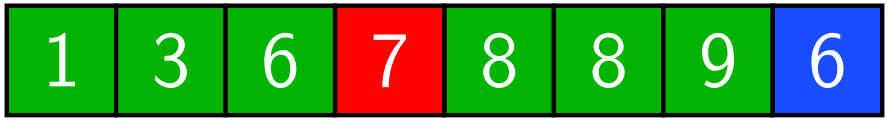
$j = 7$

Insertion Sort - Average Case



$j = 7$

Insertion Sort - Average Case



$$j = 7$$

$$t_j \approx \frac{j}{2}$$