

提纲

算法的由来

算法的定义

算法的性质

算法的表示

算法的分析

算法的表示

- 如何表示一个算法？

算法的表示

- 如何表示一个算法？



算法的表示

- 如何表示一个算法?
 - 自然语言



算法的表示

- 自然语言
 - 方法优势
 - 贴近人类思维，易于理解主旨

选择排序

- 第一次遍历找到最小元素
- 第二次在剩余数组中遍历找到次小元素
- ...
- 第 n 次在剩余数组中遍历找到第 n 小元素

算法的表示

- 自然语言
 - 方法优势
 - 贴近人类思维，易于理解主旨
 - 不便之处
 - 语言描述繁琐，容易产生歧义
 - 使用了“...”等不严谨的描述

选择排序

- 第一次遍历找到最小元素
- 第二次在剩余数组中遍历找到次小元素
- ...
- 第 n 次在剩余数组中遍历找到第 n 小元素

算法的表示

- 如何表示一个算法?

- 自然语言
- 编程语言



算法的表示

- 编程语言
 - 方法优势
 - 精准表达逻辑，规避表述歧义

选择排序

```
void select_sort(int*a,int n)
{
    int i,j,t;
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++){
            if(a[i] > a[j]){
                t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
        }
    }
}
```

C语言

```
def select_sort(a, n):
    for i in range(0, n-1):
        for j in range(i+1,n):
            if a[i] > a[j]:
                tem = a[i]
                a[i] = a[j]
                a[j] = tem
```

Python语言

算法的表示

- 编程语言

- 方法优势

- 精准表达逻辑，规避表述歧义

- 不便之处

- 不同编程语言间语法存在差异

- 过于关注算法实现的细枝末节

选择排序

```
void select_sort(int*a,int n)
{
    int i,j,t;
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++){
            if(a[i] > a[j]){
                t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
        }
    }
}
```

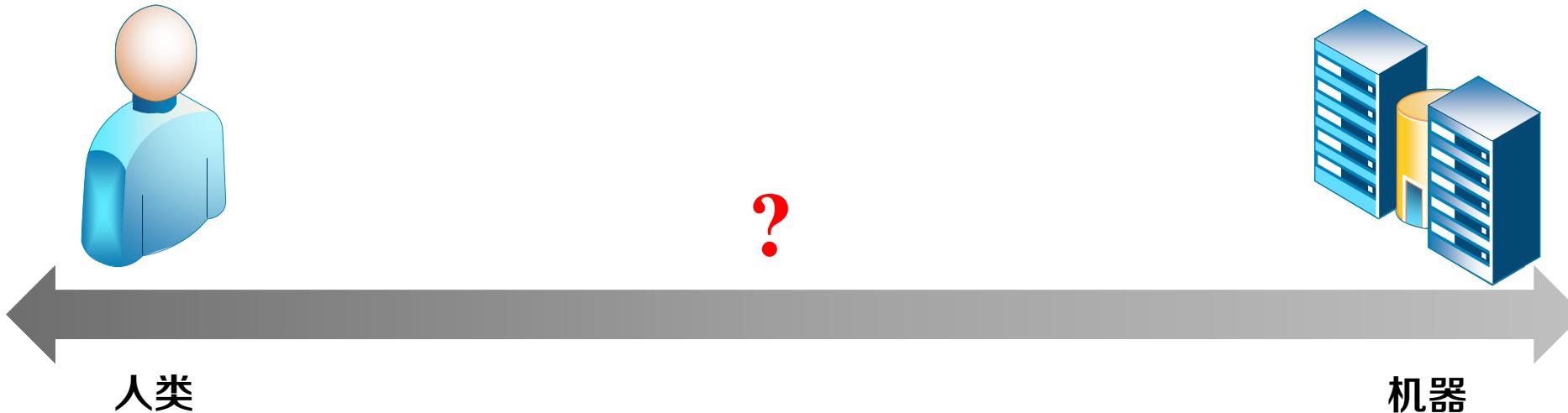
C语言

```
def select_sort(a, n):
    for i in range(0, n-1):
        for j in range(i+1,n):
            if a[i] > a[j]:
                tem = a[i]
                a[i] = a[j]
                a[j] = tem
```

Python语言

算法的表示

- 如何表示一个算法?
 - 自然语言: 贴近人类思维, 易于理解主旨
 - 编程语言: 精准表达逻辑, 规避表述歧义



问题: 可否同时兼顾两类表示方法的优势?

算法的表示

- 如何表示一个算法?
 - 自然语言: 贴近人类思维, 易于理解主旨
 - 编程语言: 精准表达逻辑, 规避表述歧义



问题: 可否同时兼顾两类表示方法的优势?

算法的表示

- 伪代码

- 非正式语言

- 移植**编程语言**书写形式作为基础和框架
 - 按照接近**自然语言**的形式表达算法过程

选择排序

输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

```
for i ← 1 to n – 1 do
    for j ← i + 1 to n do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

算法的表示

● 伪代码

- 非正式语言

- 移植**编程语言**书写形式作为基础和框架
 - 按照接近**自然语言**的形式表达算法过程

- 兼顾自然语言与编程语言优势

- **简洁**表达算法本质，不拘泥于实现细节
 - **准确**反映算法过程，不产生矛盾和歧义

选择排序

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

算法的表示

- 伪代码

- 书写约定

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

选择排序

算法的表示

- 伪代码

- 书写约定

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

定义算法的输入和输出

选择排序

算法的表示

- 伪代码

- 书写约定

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

循环语句块缩进

选择排序

算法的表示

- 伪代码

- 书写约定

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

将 $i + 1$ 赋值给j

选择排序

算法的表示

- 伪代码

- 书写约定

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

注释使用“//”符号

选择排序

算法的表示

- 伪代码

- 书写约定

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

条件语句块缩进

选择排序

算法的表示

- 伪代码

- 书写约定

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

不关注交换过程的实现细节

选择排序

算法的表示

- 伪代码
 - 书写约定

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

选择排序

之后出现的算法均使用伪代码描述

算法的表示

- 伪代码

- 示例解读

24	17	40	28	13	14	22	32	40	21	48	4	47	8	37	18
----	----	----	----	----	----	----	----	----	----	----	---	----	---	----	----

输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $i \leftarrow 1$ to $n - 1$ do

 for $j \leftarrow i + 1$ to n do

 //如果 $A[i]$ 大于 $A[j]$, 则交换二者位置

 if $A[i] > A[j]$ then

 | 交换 $A[i]$ 和 $A[j]$

 end

 end

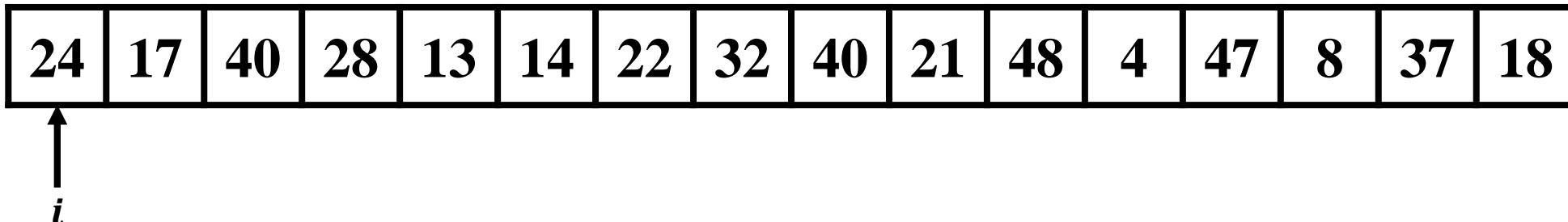
end

选择排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

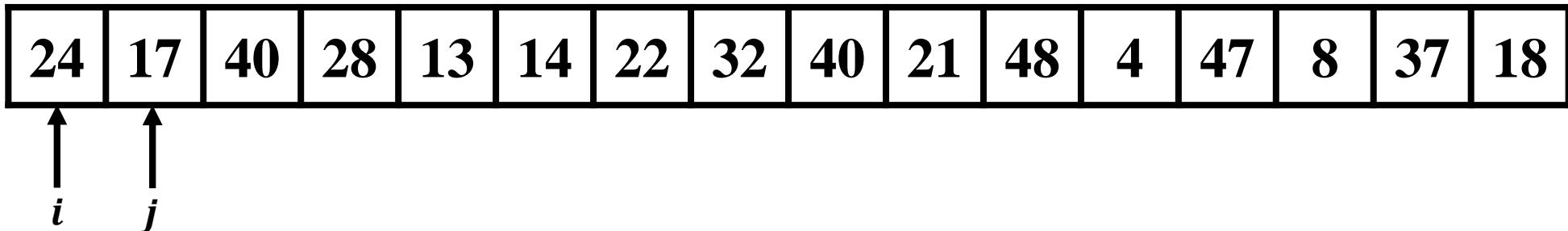
从数组首部开始枚举*i*

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$
输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

```
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

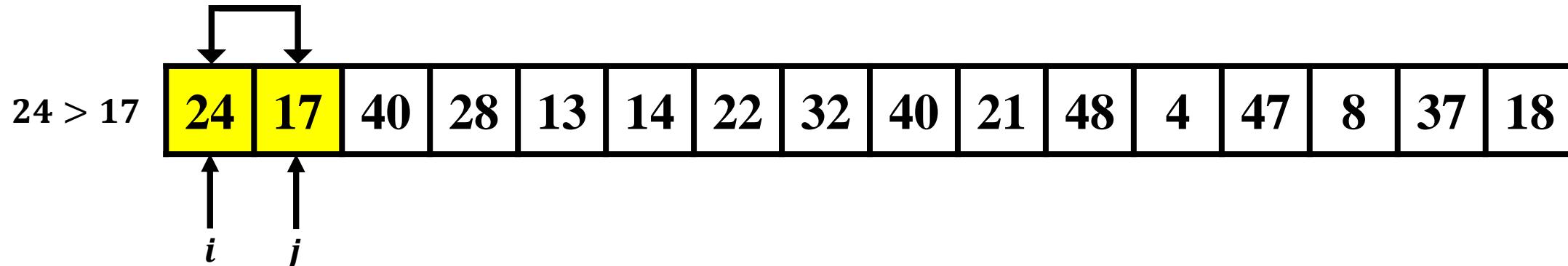
从 $i + 1$ 开始枚举 j

选择排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

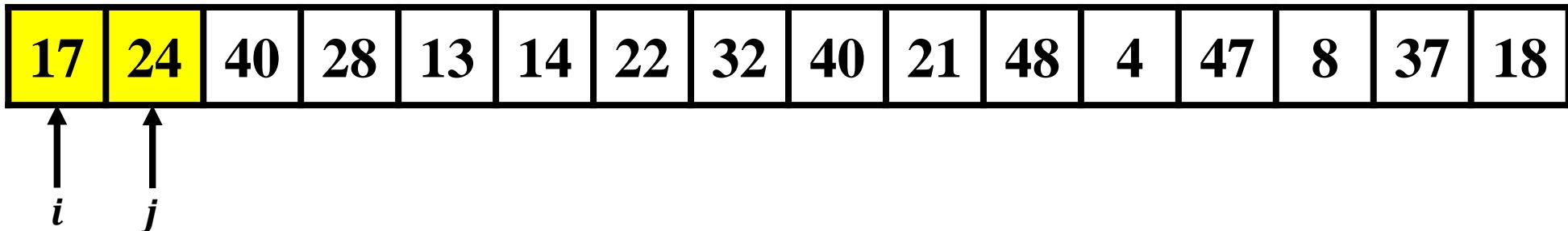
当第 i 个元素大于第 j 个元素时,
交换元素位置

选择排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

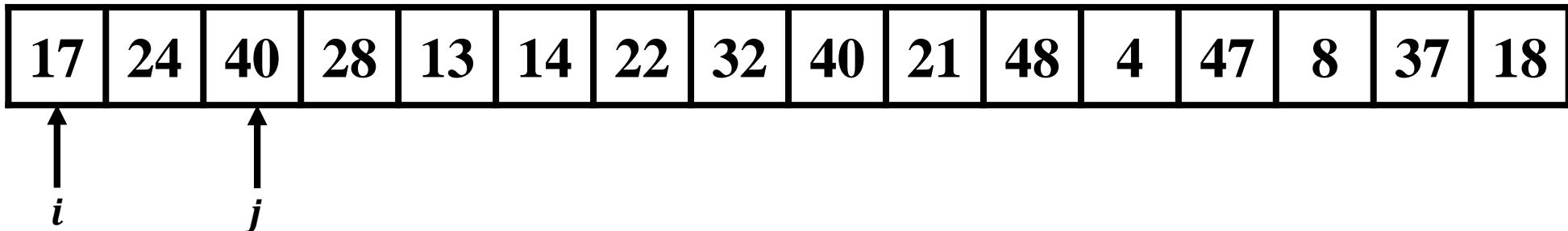
当第 i 个元素大于第 j 个元素时,
交换元素位置

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$
输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

```
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

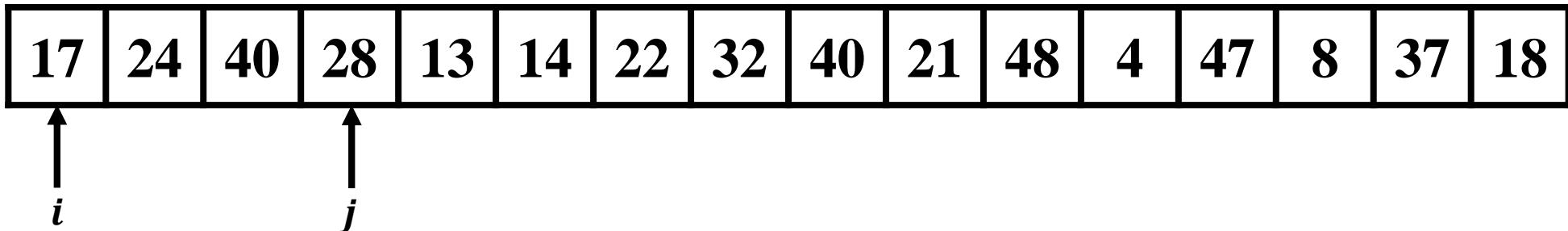
继续枚举 j

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$
输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $i \leftarrow 1$ to $n - 1$ do

 for $j \leftarrow i + 1$ to n do

 //如果 $A[i]$ 大于 $A[j]$, 则交换二者位置

 if $A[i] > A[j]$ then

 | 交换 $A[i]$ 和 $A[j]$

 end

 end

end

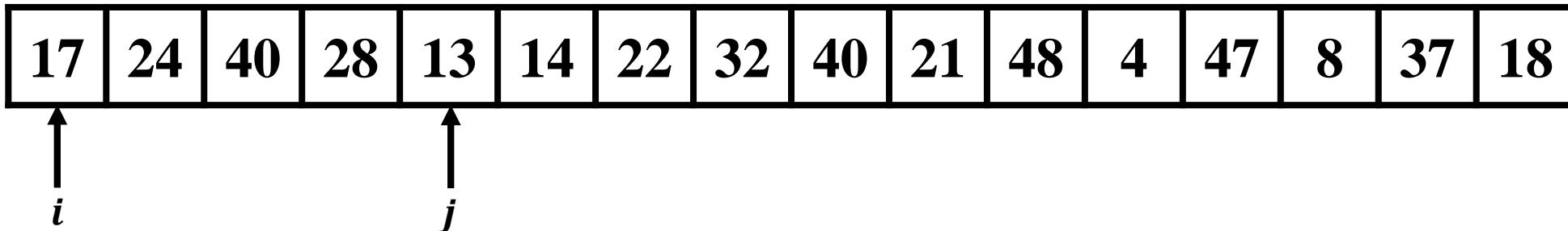
继续枚举 *j*

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$
输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

```
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

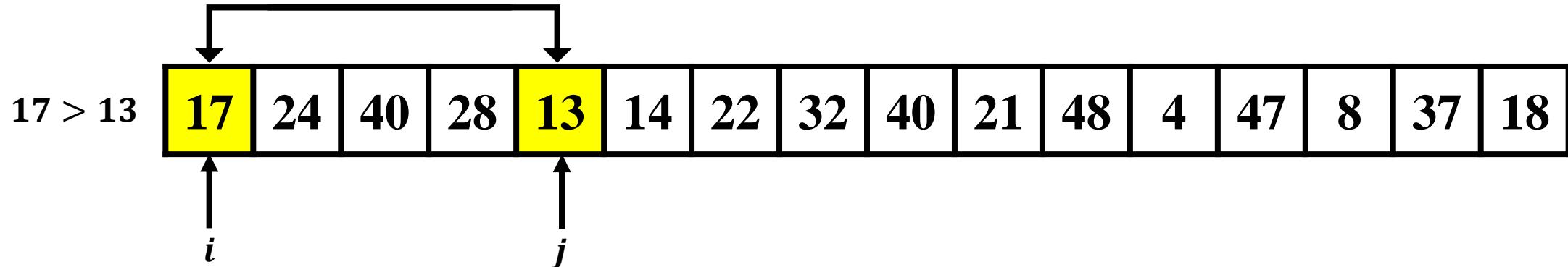
继续枚举 *j*

选择排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

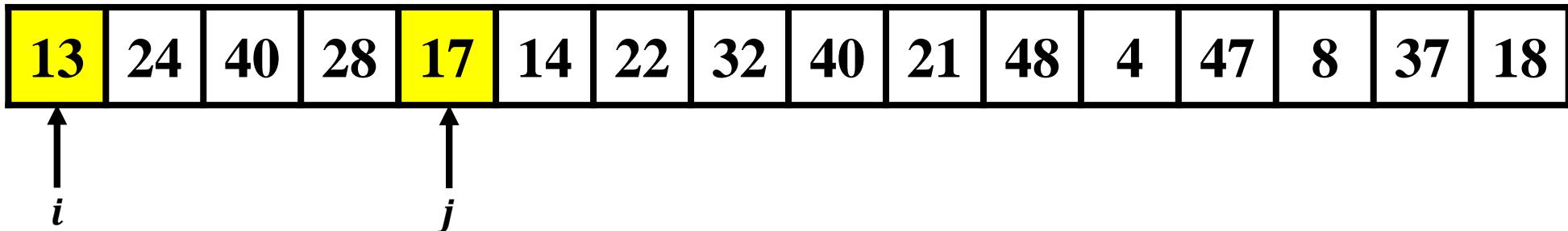
当第*i*个元素大于第*j*个元素时,
交换元素位置

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $i \leftarrow 1$ to $n - 1$ do

 for $j \leftarrow i + 1$ to n do

 //如果 $A[i]$ 大于 $A[j]$, 则交换二者位置

 if $A[i] > A[j]$ then

 | 交换 $A[i]$ 和 $A[j]$

 end

 end

end

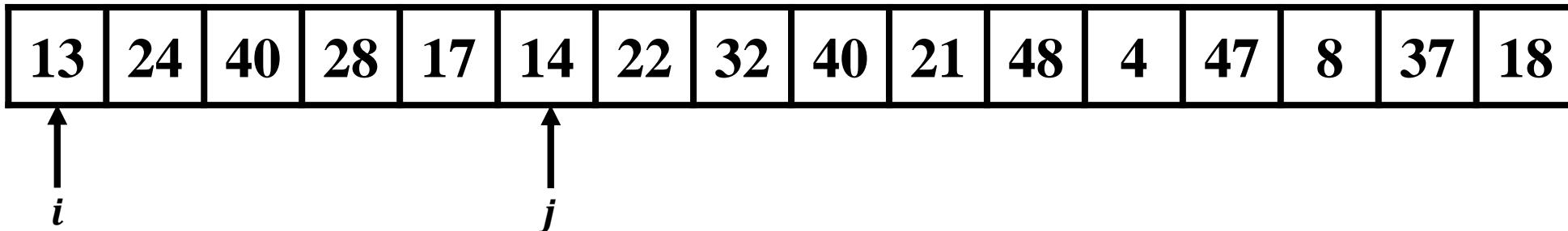
当第*i*个元素大于第*j*个元素时,
交换元素位置

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$
输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

```
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

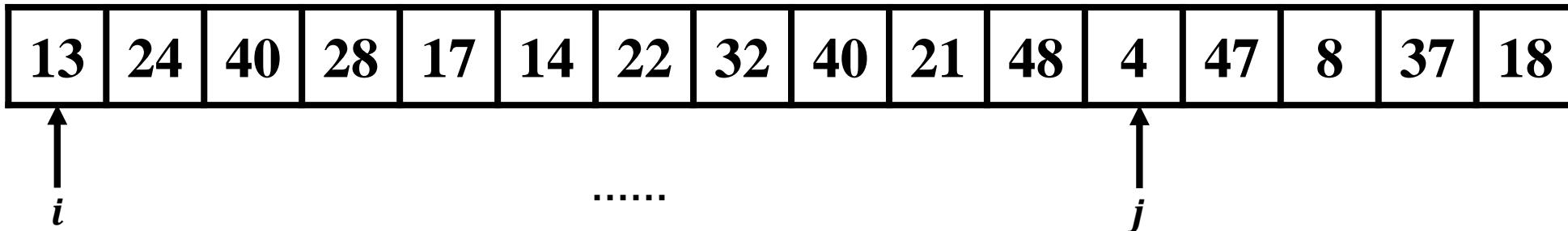
继续枚举 *j*

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$
输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $i \leftarrow 1$ to $n - 1$ do

 for $j \leftarrow i + 1$ to n do

 //如果 $A[i]$ 大于 $A[j]$, 则交换二者位置

 if $A[i] > A[j]$ then

 | 交换 $A[i]$ 和 $A[j]$

 end

 end

end

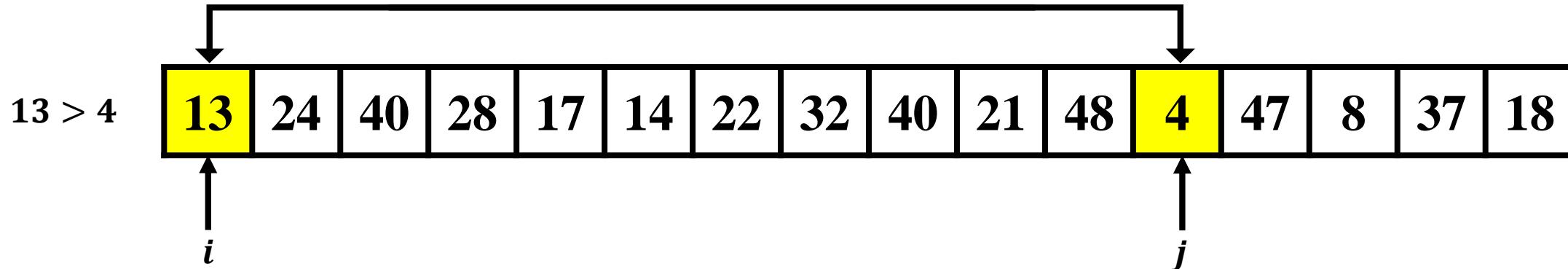
继续枚举 j

选择排序

算法的表示

● 伪代码

● 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A' [a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $i \leftarrow 1$ **to** $n - 1$ **do**

for $j \leftarrow i + 1$ **to** n **do**

//如果 $A[i]$ 大于 $A[j]$ ，则交换二者位置

if $A[i] > A[j]$ **then**

| 交换 $A[i]$ 和 $A[j]$

-
end

end

end

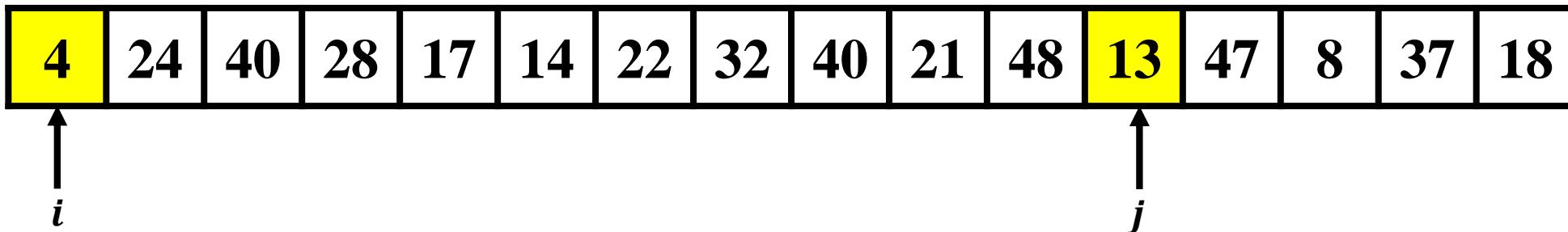
当第*i*个元素大于第*j*个元素时，
交换元素位置

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $i \leftarrow 1$ to $n - 1$ do

 for $j \leftarrow i + 1$ to n do

 //如果 $A[i]$ 大于 $A[j]$, 则交换二者位置

 if $A[i] > A[j]$ then

 | 交换 $A[i]$ 和 $A[j]$

 end

 end

end

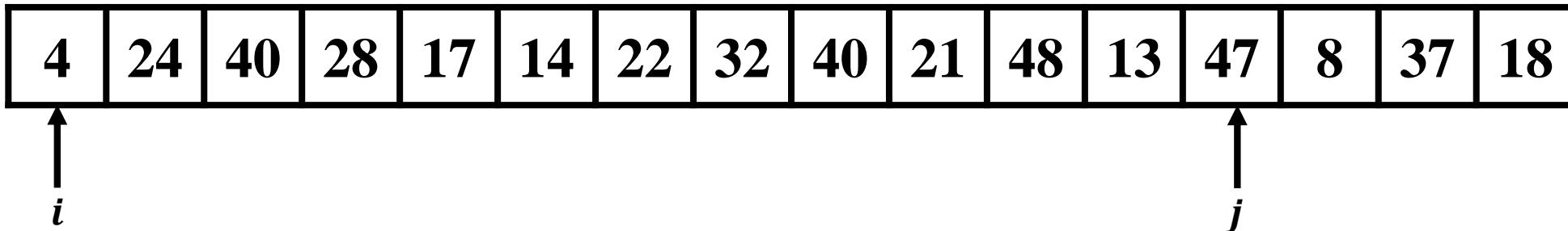
当第*i*个元素大于第*j*个元素时,
交换元素位置

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$
输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

```
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

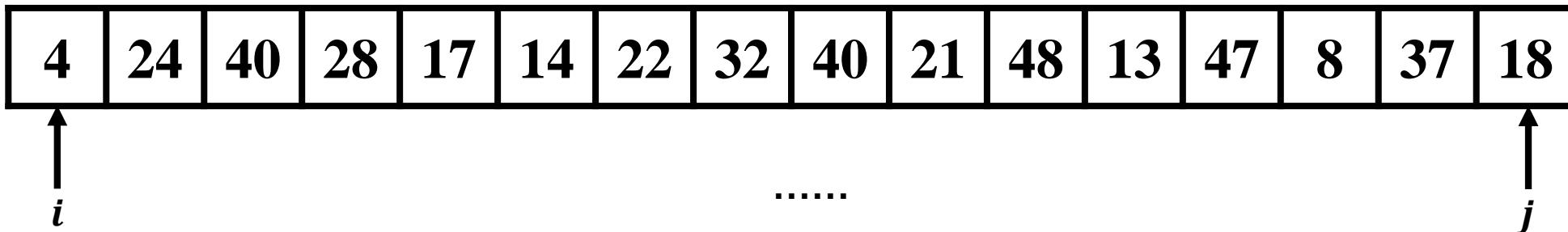
继续枚举 *j*

选择排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$
输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

```
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

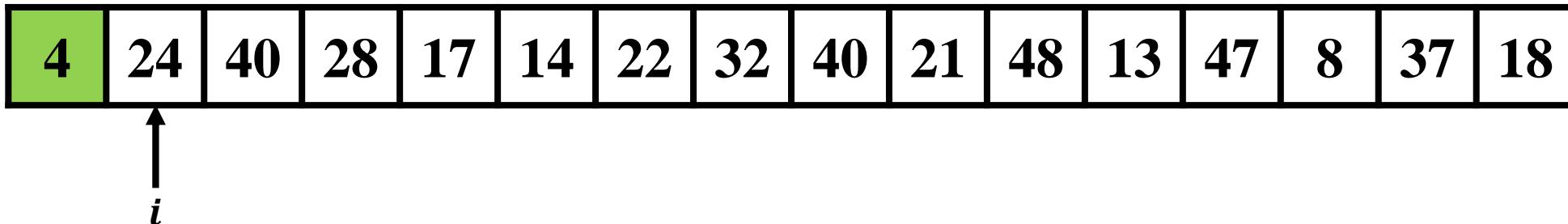
枚举 j 至 n , 结束循环

选择排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

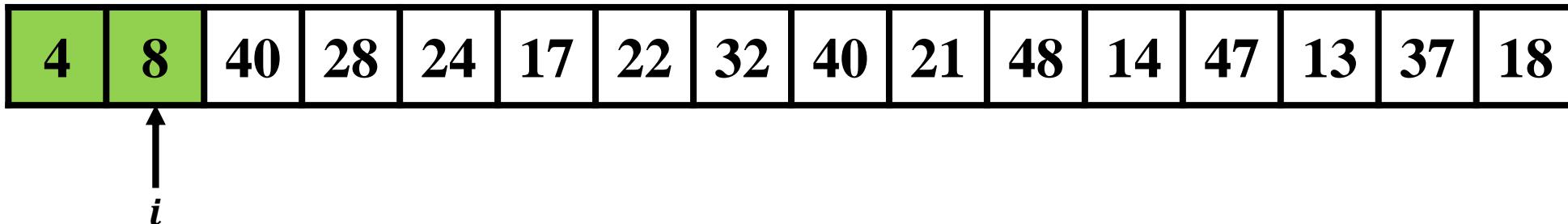
继续枚举*i*

选择排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

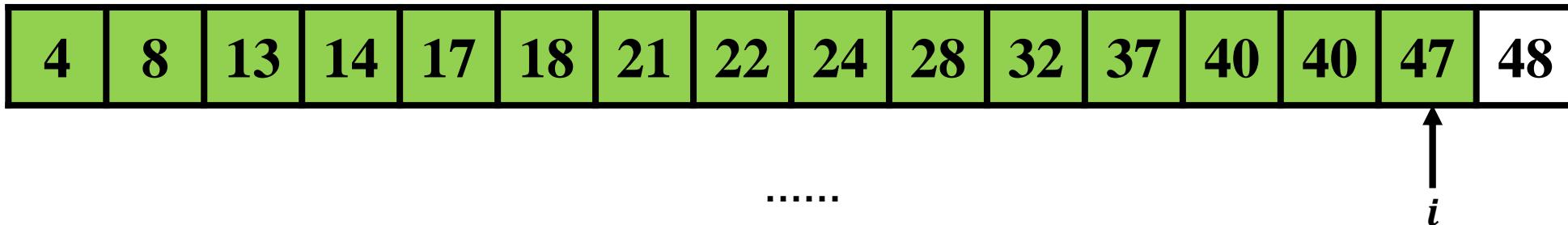
继续枚举*i*

选择排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
        //如果  $A[i]$  大于  $A[j]$ , 则交换二者位置
        if  $A[i] > A[j]$  then
            | 交换  $A[i]$  和  $A[j]$ 
        end
    end
end
```

枚举*i*至*n* – 1, 结束循环

选择排序

算法的表示

- 伪代码

- 示例解读

4	8	13	14	17	18	21	22	24	28	32	37	40	40	47	48
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $i \leftarrow 1$ to $n - 1$ do

 for $j \leftarrow i + 1$ to n do

 //如果 $A[i]$ 大于 $A[j]$, 则交换二者位置

 if $A[i] > A[j]$ then

 | 交换 $A[i]$ 和 $A[j]$

 end

 end

end

选择排序

算法的表示

- 伪代码

- 示例解读

17	24	28	40	13	14	22	32	40	21	48	4	47	8	37	18
----	----	----	----	----	----	----	----	----	----	----	---	----	---	----	----

输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

 //将 $A[j]$ 插入到已排序的数组 $A[1..j - 1]$ 中

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

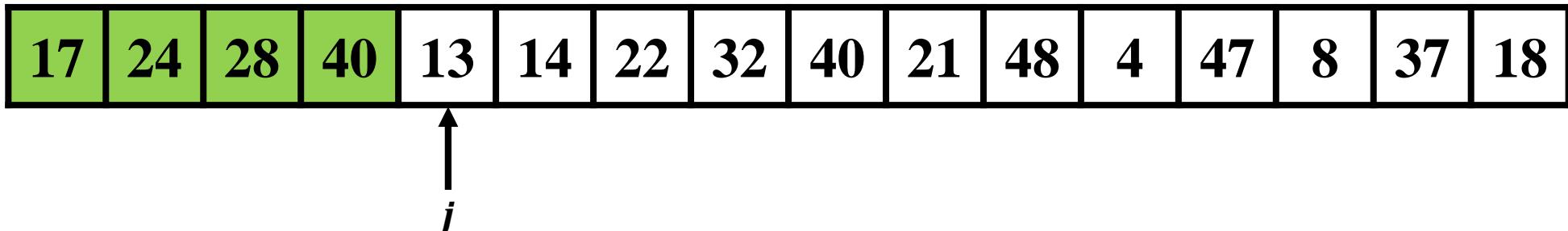
插入排序

算法的表示

- 伪代码

- 示例解读

$key = 13$



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

 //将 $A[j]$ 插入到已排序的数组 $A[1..j - 1]$ 中

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

将 $A[j]$ 赋值给 key

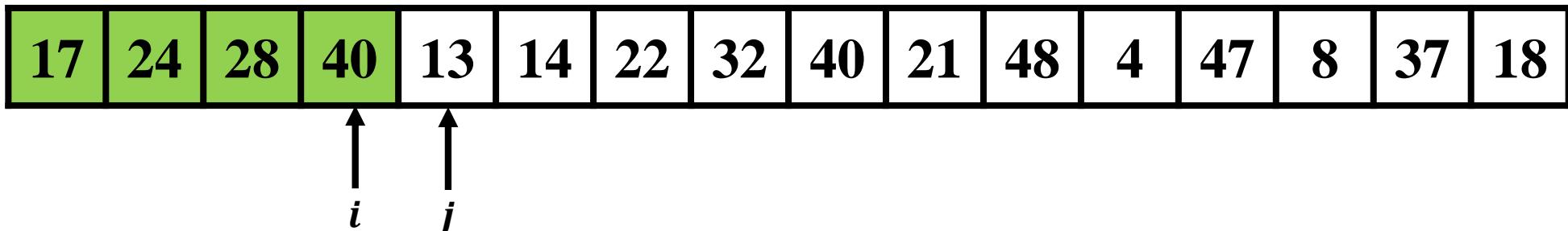
插入排序

算法的表示

- 伪代码

- 示例解读

$key = 13$



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

 //将 $A[j]$ 插入到已排序的数组 $A[1..j - 1]$ 中

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

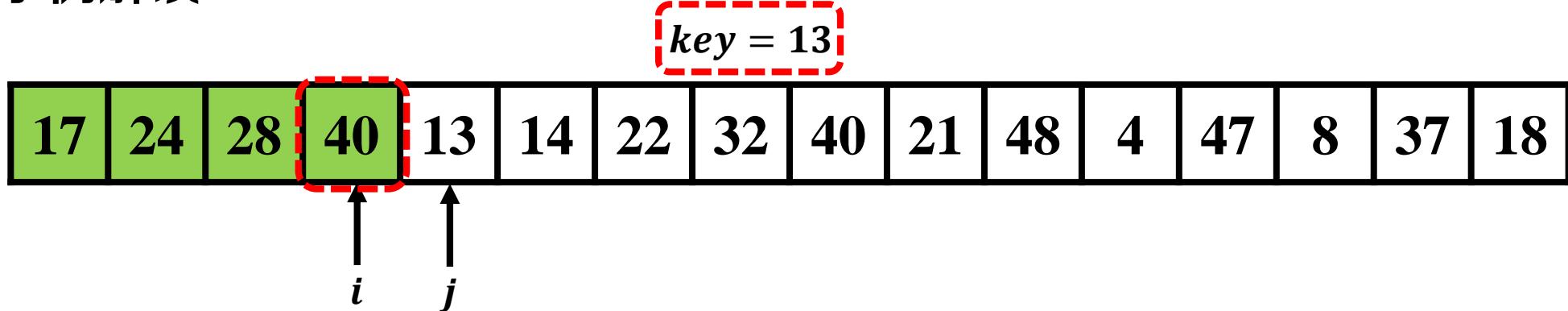
从 $j - 1$ 开始枚举 i

插入排序

算法的表示

- 伪代码

- 示例解读



$i > 0$
 $A[i] > key$

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

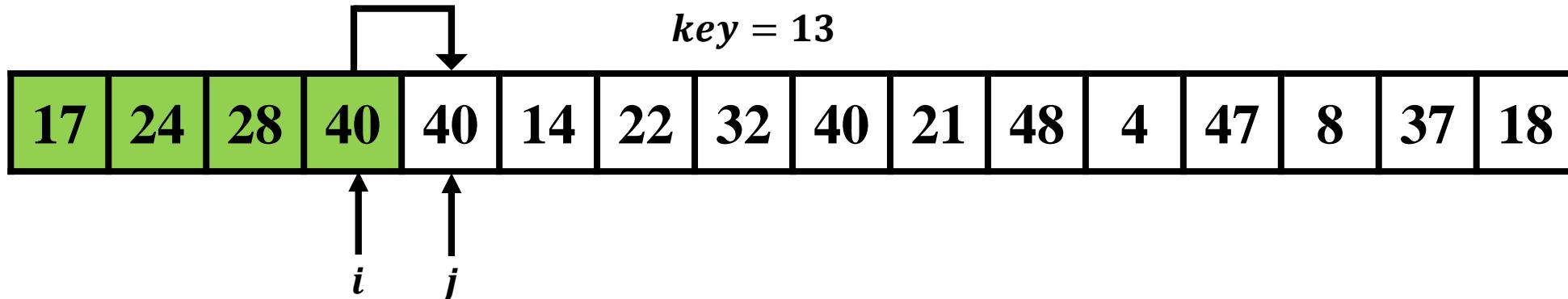
判断循环条件

插入排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

 //将 $A[j]$ 插入到已排序的数组 $A[1..j - 1]$ 中

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

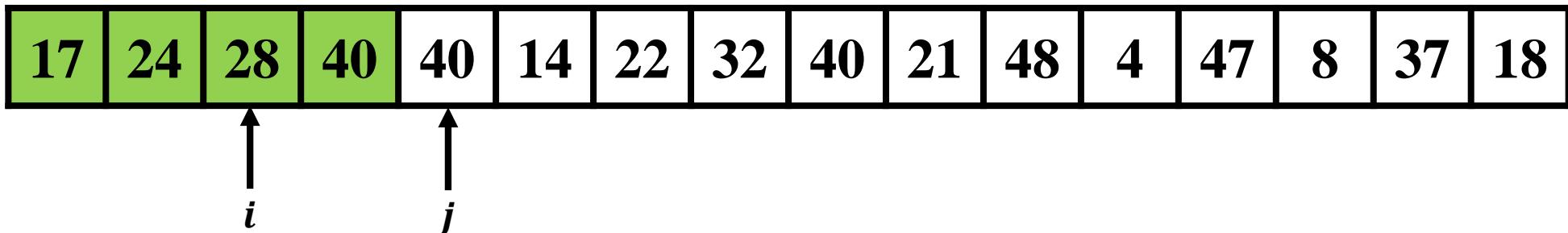
将 $A[i]$ 赋值给 $A[i + 1]$

算法的表示

- 伪代码

- 示例解读

$key = 13$



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

 //将 $A[j]$ 插入到已排序的数组 $A[1..j - 1]$ 中

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

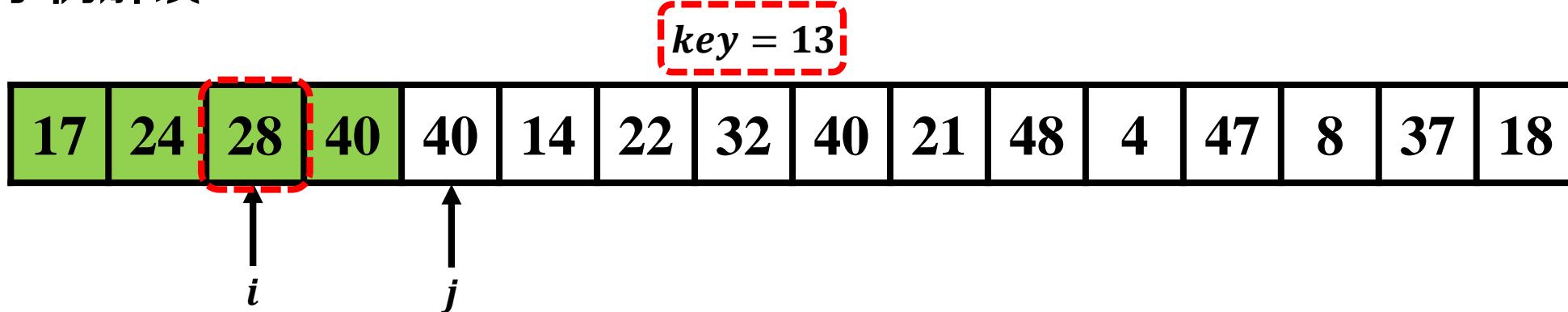
将 i 左移一步

插入排序

算法的表示

- 伪代码

- 示例解读



$i > 0$
 $A[i] > key$

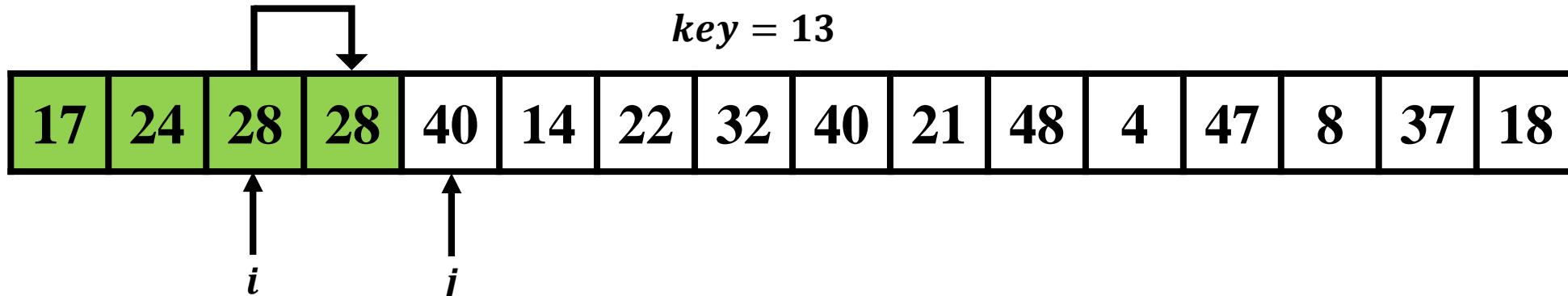
```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

判断循环条件

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

 //将 $A[j]$ 插入到已排序的数组 $A[1..j - 1]$ 中

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$\boxed{A[i + 1] \leftarrow A[i]}$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

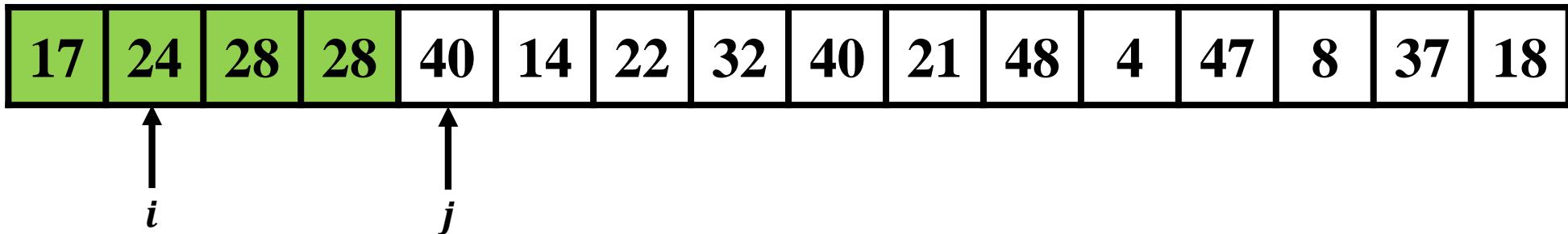
将 $A[i]$ 赋值给 $A[i + 1]$

算法的表示

- 伪代码

- 示例解读

$key = 13$



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

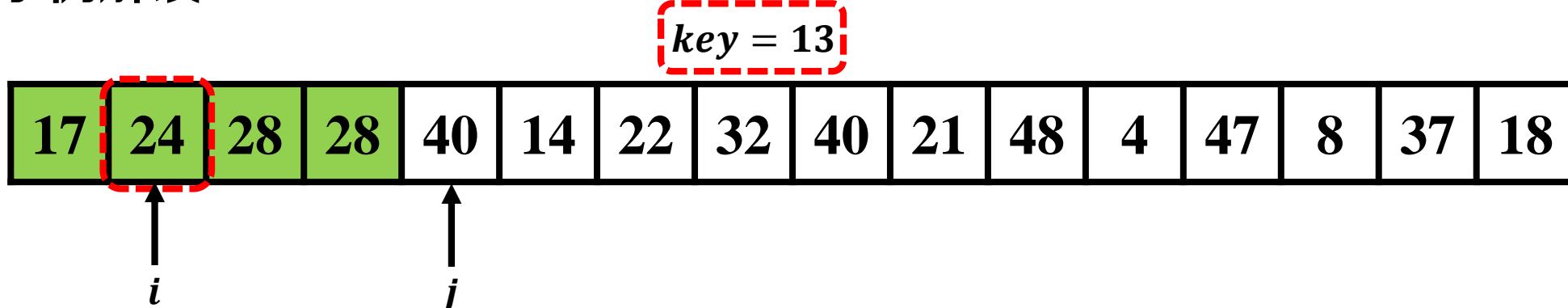
将 i 左移一步

插入排序

算法的表示

- 伪代码

- 示例解读



$i > 0$
 $A[i] > key$

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

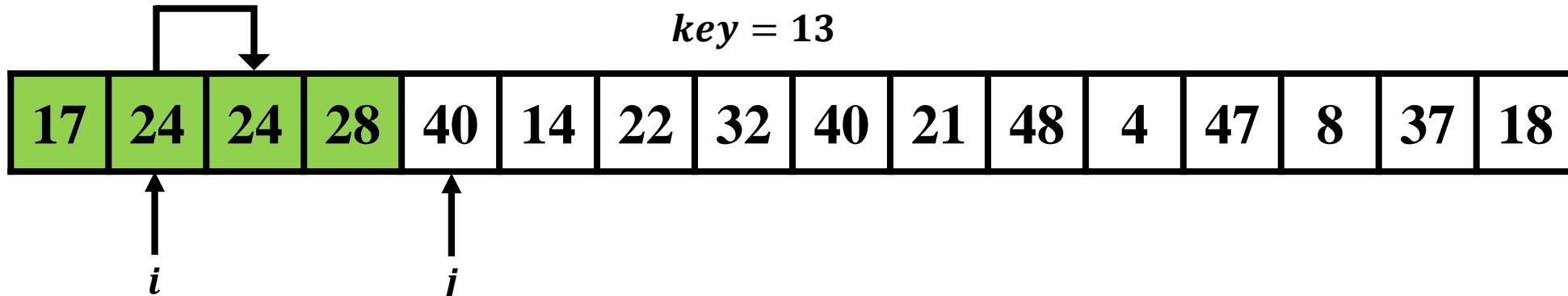
判断循环条件

插入排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

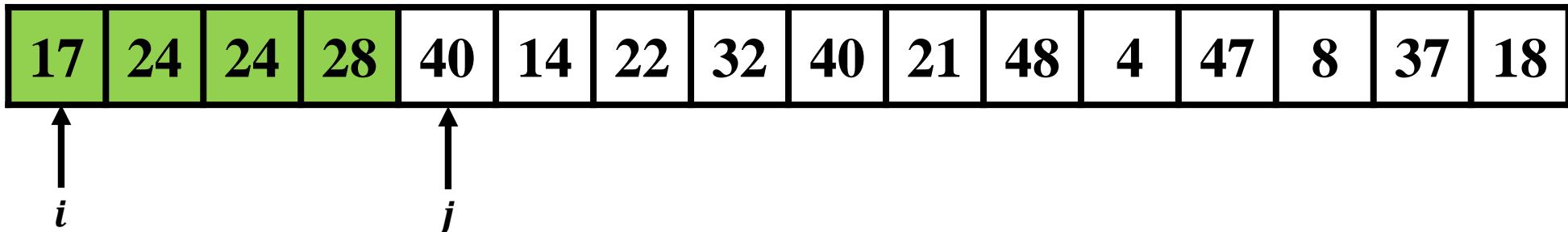
将 $A[i]$ 赋值给 $A[i + 1]$

算法的表示

- 伪代码

- 示例解读

$key = 13$



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

 //将 $A[j]$ 插入到已排序的数组 $A[1..j - 1]$ 中

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

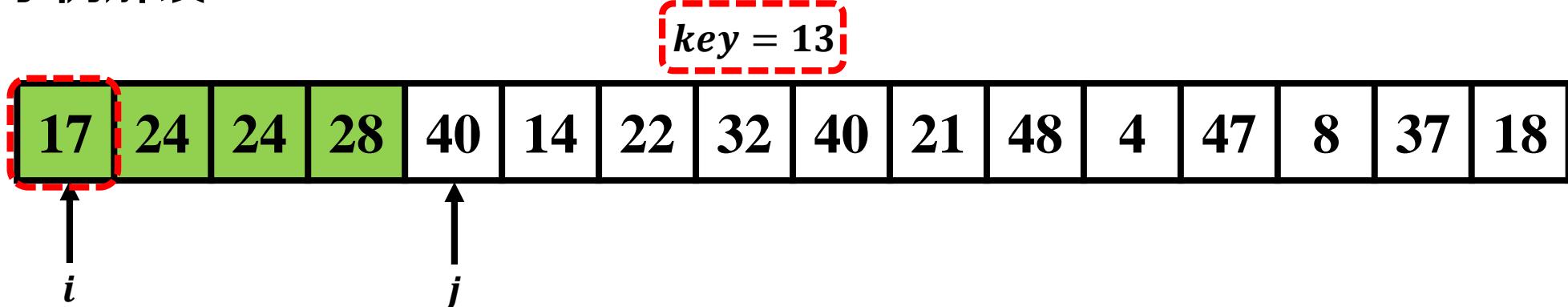
将 i 左移一步

插入排序

算法的表示

- 伪代码

- 示例解读



$i > 0$
 $A[i] > key$

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j-1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

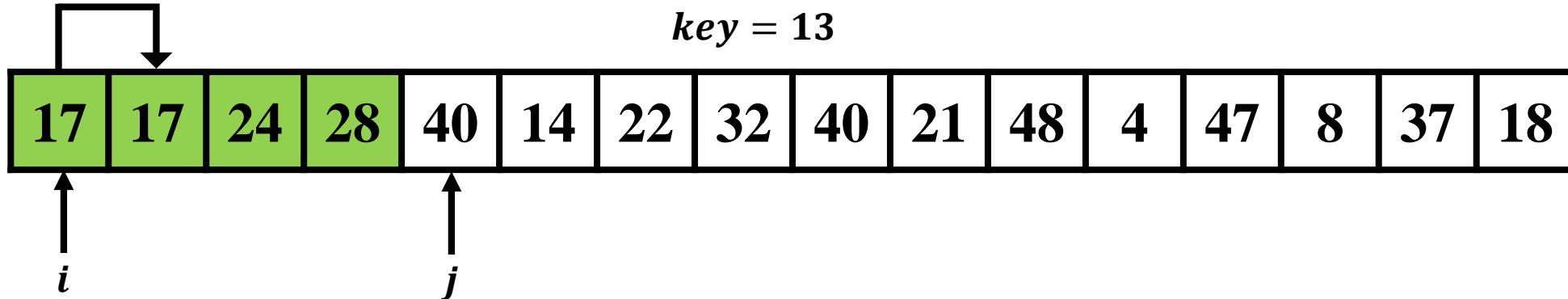
判断循环条件

插入排序

算法的表示

- 伪代码

- 示例解读



输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

 //将 $A[j]$ 插入到已排序的数组 $A[1..j - 1]$ 中

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

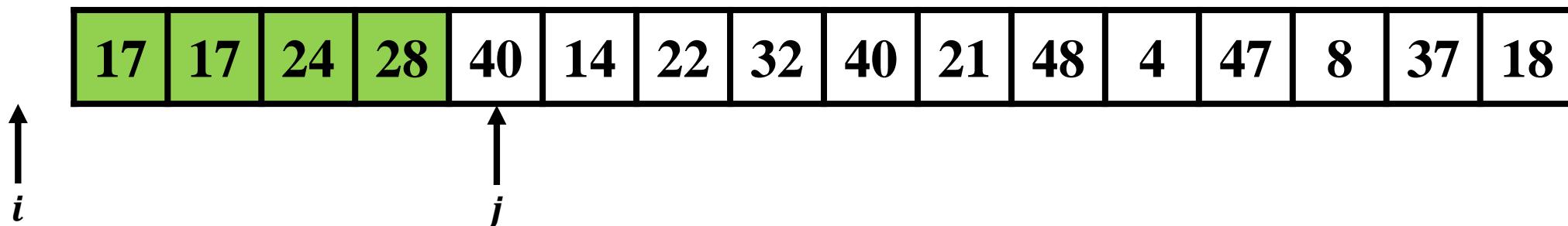
将 $A[i]$ 赋值给 $A[i + 1]$

算法的表示

- 伪代码

- 示例解读

$key = 13$



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

将 i 左移一步

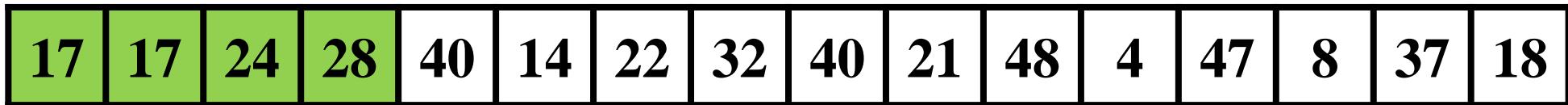
插入排序

算法的表示

- 伪代码

- 示例解读

$key = 13$



i

j

$i = 0$
不满足循环条件

```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

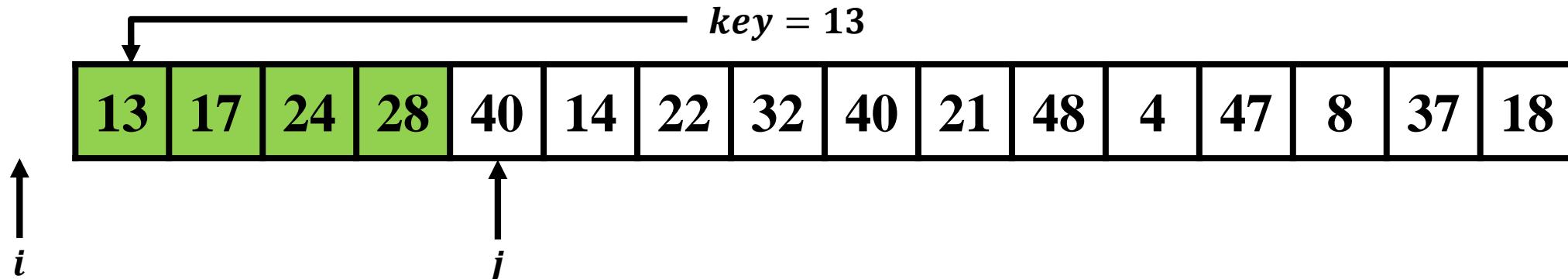
判断循环条件

插入排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
    key  $\leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > \text{key}$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
    A[i + 1]  $\leftarrow \text{key}$ 
end
```

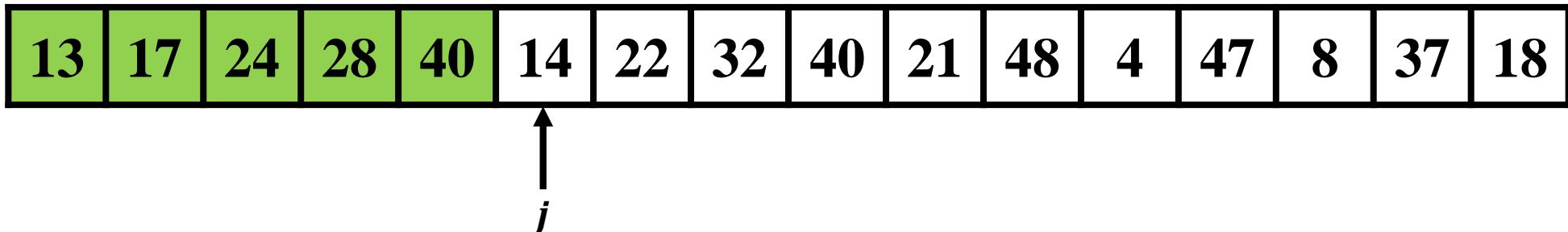
将 *key* 赋值给 $A[i + 1]$

插入排序

算法的表示

- 伪代码

- 示例解读



```
输入: 数组  $A[a_1, a_2, \dots, a_n]$ 
输出: 升序数组  $A'[a'_1, a'_2, \dots, a'_n]$ , 满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 
for  $j \leftarrow 2$  to  $n$  do
    key  $\leftarrow A[j]$ 
    //将  $A[j]$  插入到已排序的数组  $A[1..j - 1]$  中
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > \text{key}$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow \text{key}$ 
end
```

插入排序

算法的表示方式比较

表示方式	语言特点
自然语言	贴近人类思维，易于理解主旨 表述不够精准，存在模糊歧义
编程语言	精准表达逻辑，规避表述歧义 受限语法细节，增大理解难度
伪代码	关注算法本质，便于书写阅读