

# 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

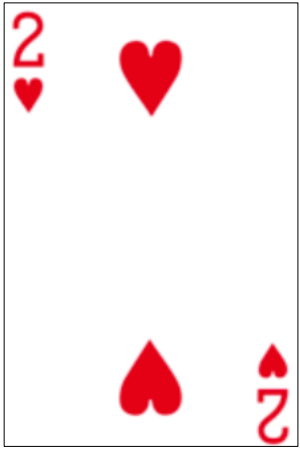
# 生活中的算法

---



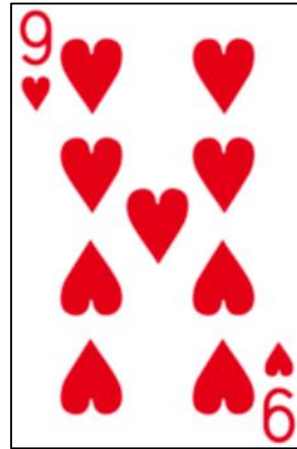
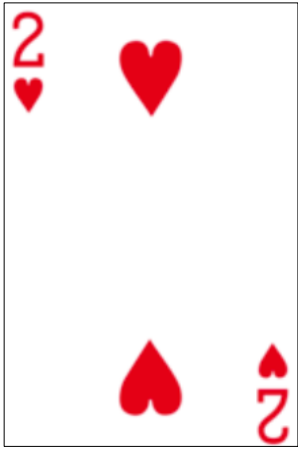
# 生活中的算法

---



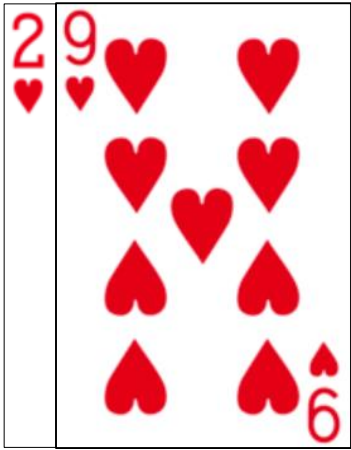
# 生活中的算法

---



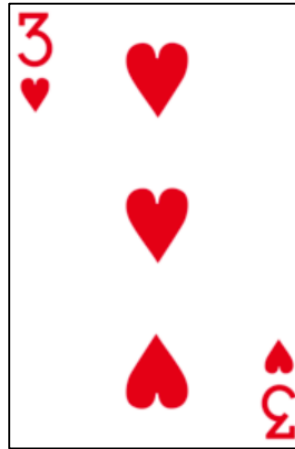
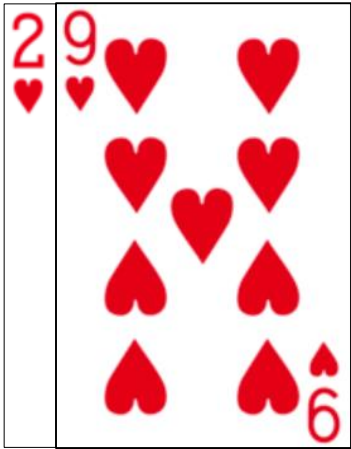
# 生活中的算法

---



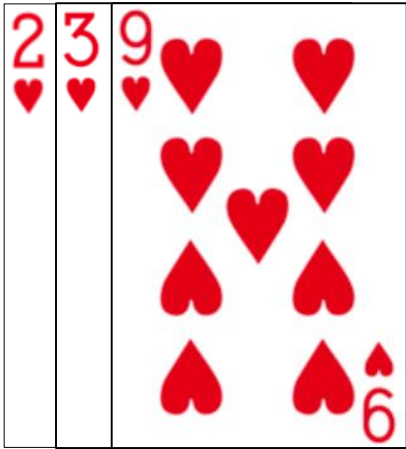
# 生活中的算法

---



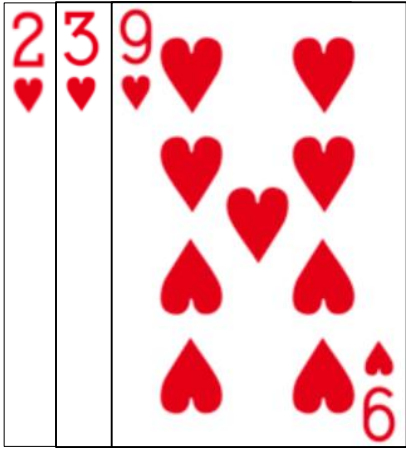
# 生活中的算法

---



# 生活中的算法

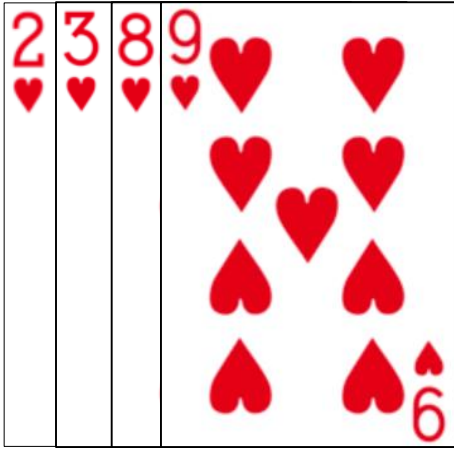
---





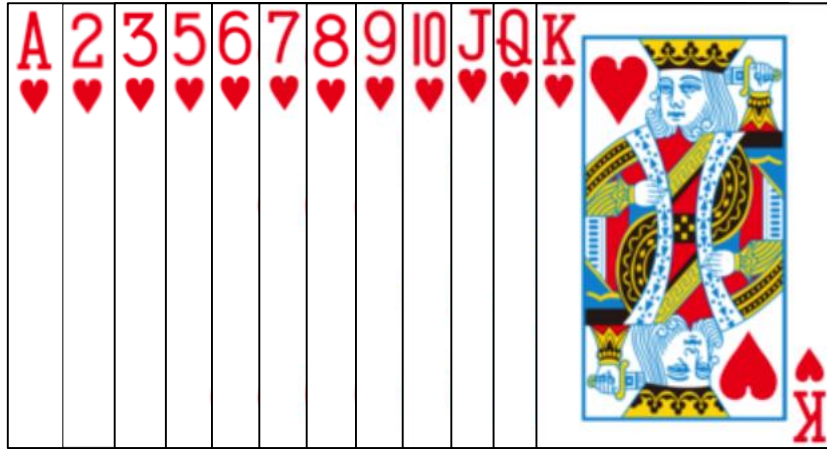
# 生活中的算法

---



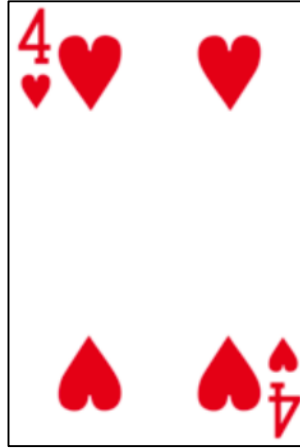
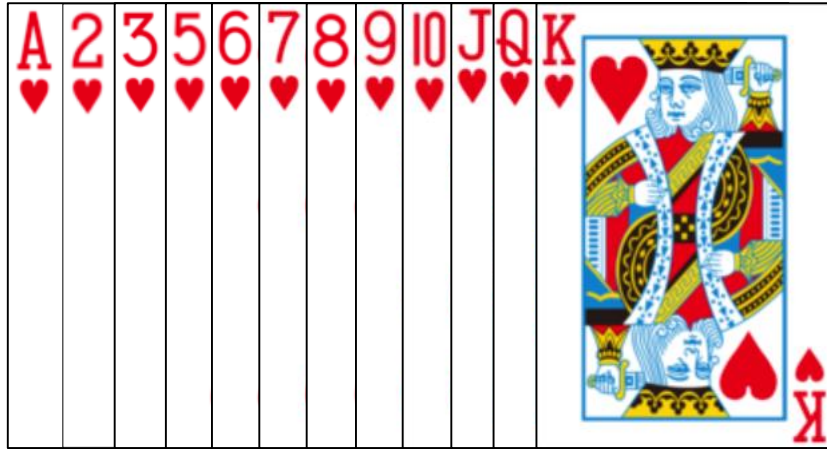
# 生活中的算法

---



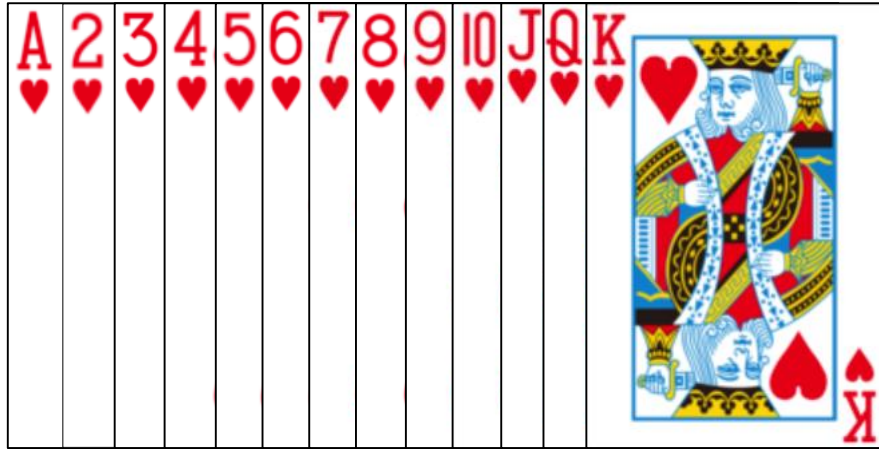
# 生活中的算法

---



# 生活中的算法

---



# 计算问题

---

- 定义
  - 给定数据输入，计算满足某种性质输出的问题

# 计算问题

---

- 定义

- 给定数据输入，计算满足某种性质输出的问题

- 示例

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 计算问题

- 定义

- 给定数据输入，计算满足某种性质输出的问题

- 示例

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

输入：  
包含16个数字的数组

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

# 计算问题

- 定义

- 给定数据输入，计算满足某种性质输出的问题

- 示例

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

输入：  
包含16个数字的数组

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

输出：  
满足升序性质的输入数组

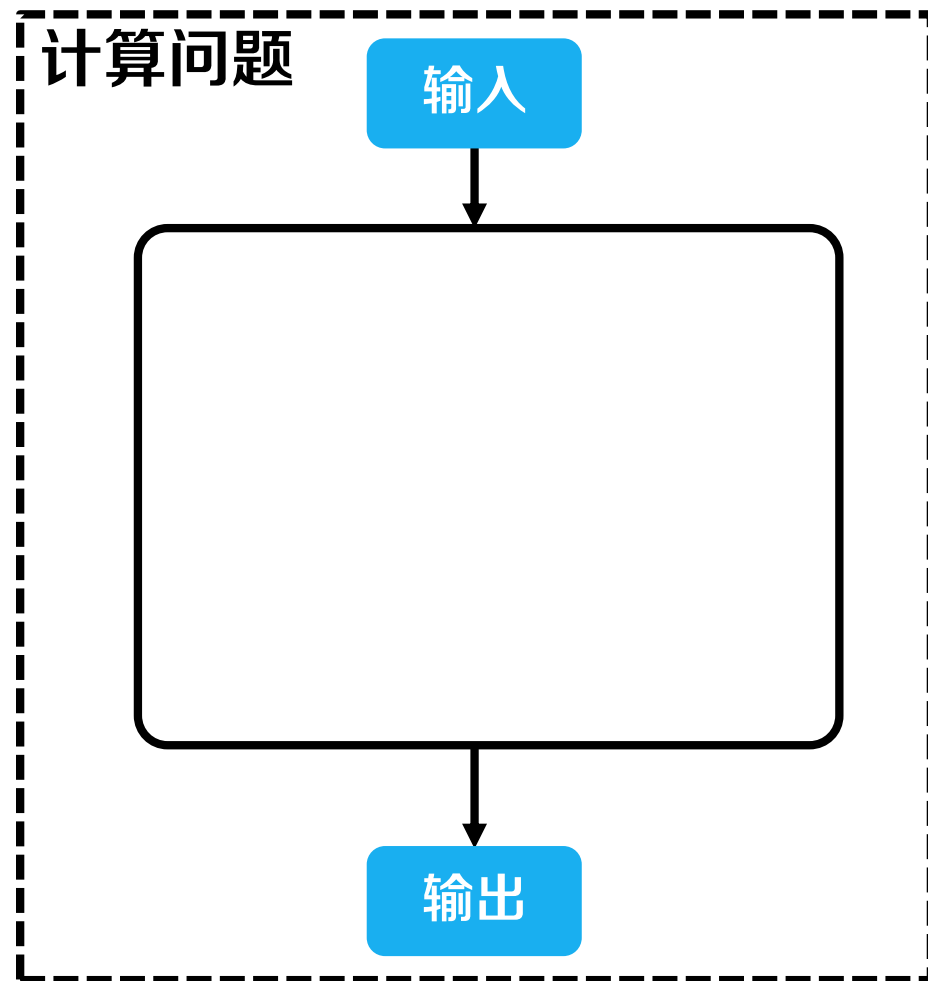
4	8	13	14	17	18	21	22	24	28	32	37	40	40	47	48
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----



# 算法的定义

---

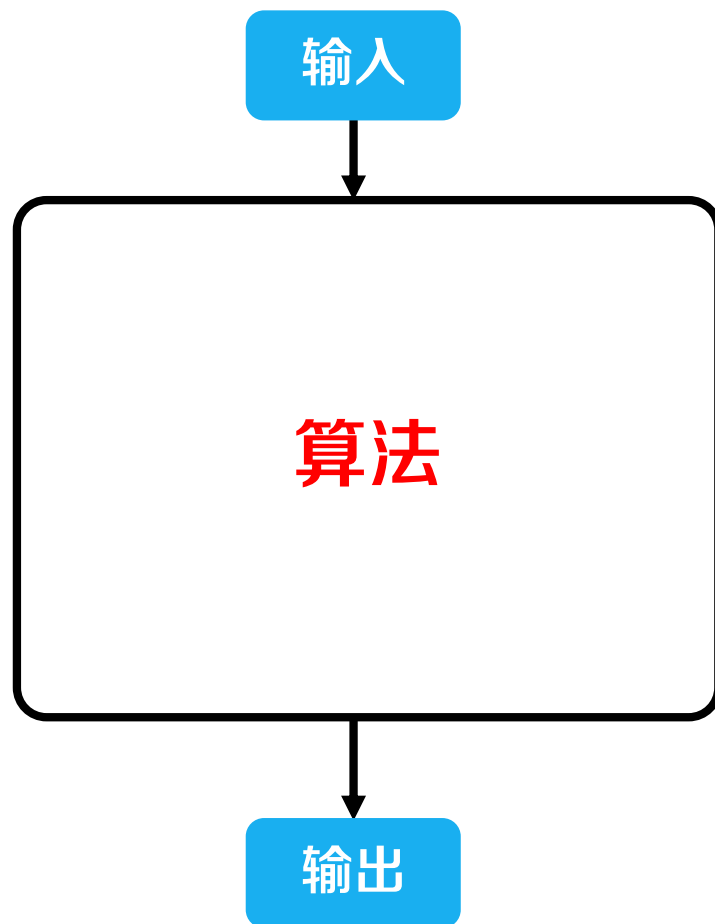
- 给定计算问题



# 算法的定义

---

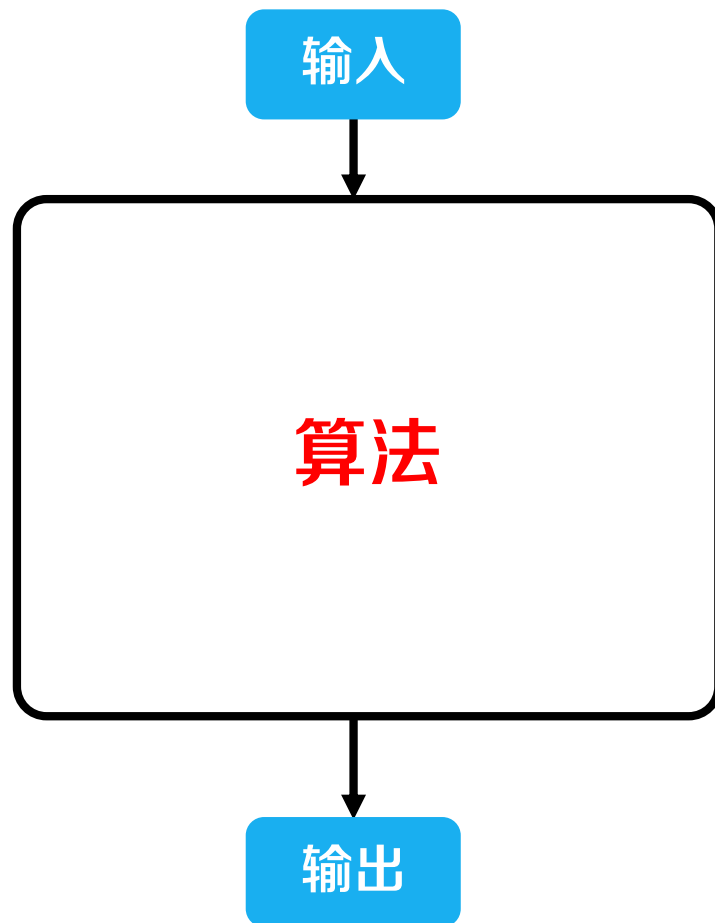
- 给定计算问题，**算法**是一系列良定义的计算步骤



# 算法的定义

- 给定计算问题，算法是一系列**良定义**的计算步骤

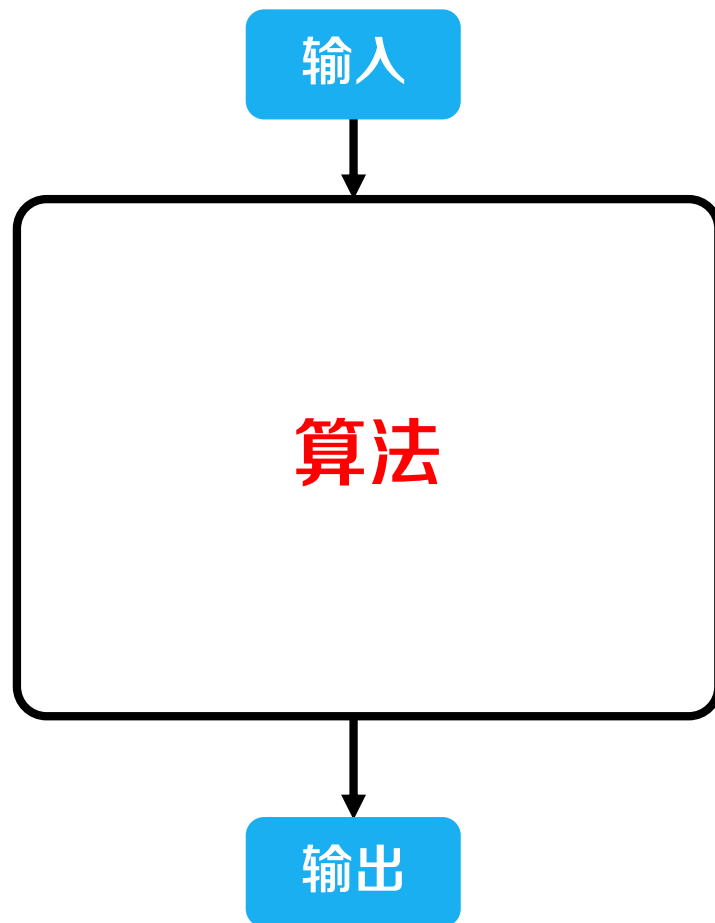
定义明确无歧义



# 算法的定义

---

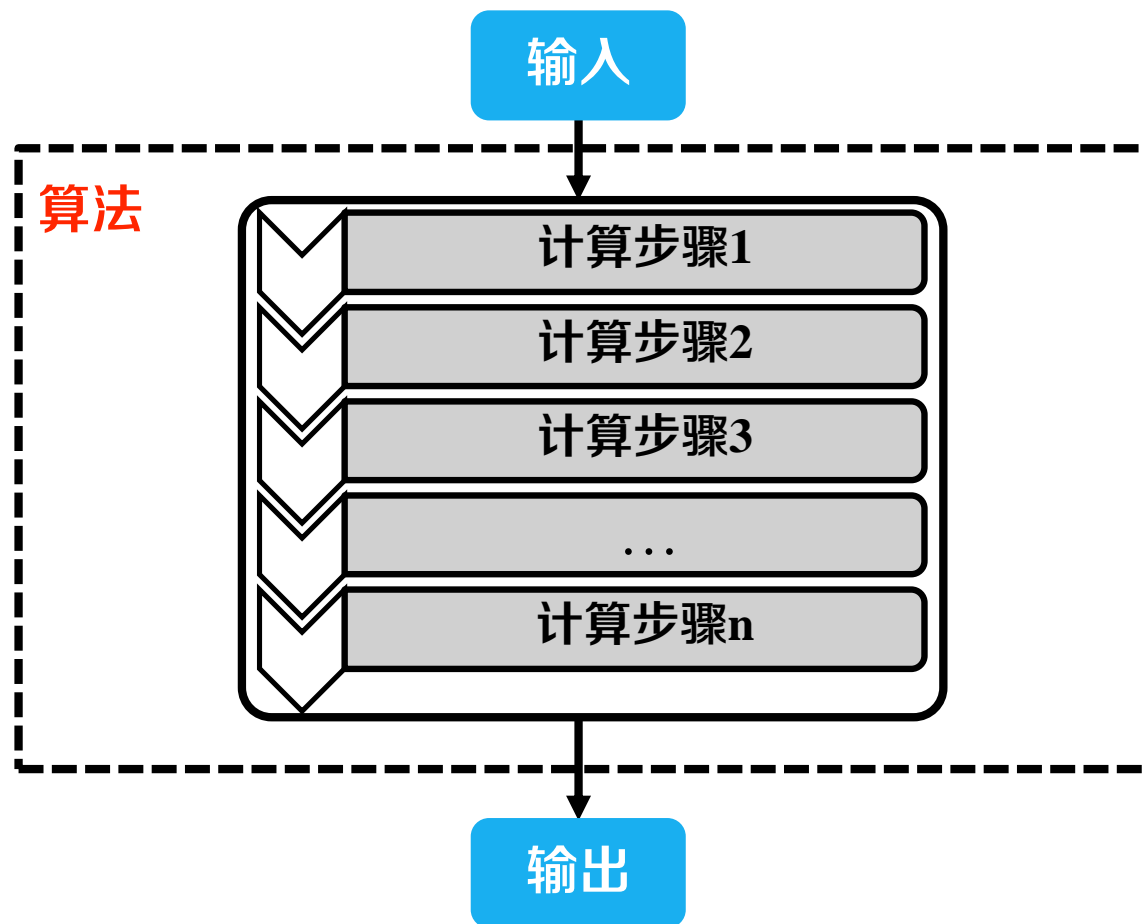
- 给定计算问题，算法是一系列良定义的**计算步骤**



计算机可实现的指令

# 算法的定义

- 给定计算问题，算法是一系列良定义的计算步骤，逐一执行计算步骤即可得预期的输出



# 插入排序

## 排序问题

### Sorting Problem

输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

满足  $a'_1 \leq a'_2 \leq \dots \leq a'_n$

- **插入排序算法**

- 将数组待排序元素依次插入到已排序部分，使已排序部分保持升序的性质

# 插入排序：算法实例

---

- 输入：

- $\langle 24, 17, 40, 28, 13, 14, 22, 32, 40, 21, 48, 4, 47, 8, 37, 18 \rangle$

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

# 插入排序：算法实例

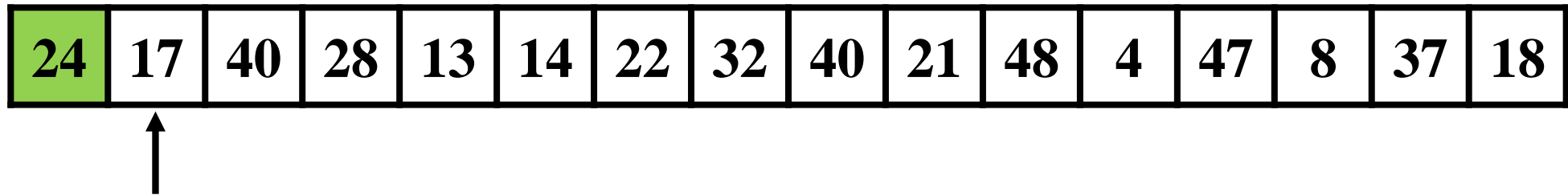
---

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



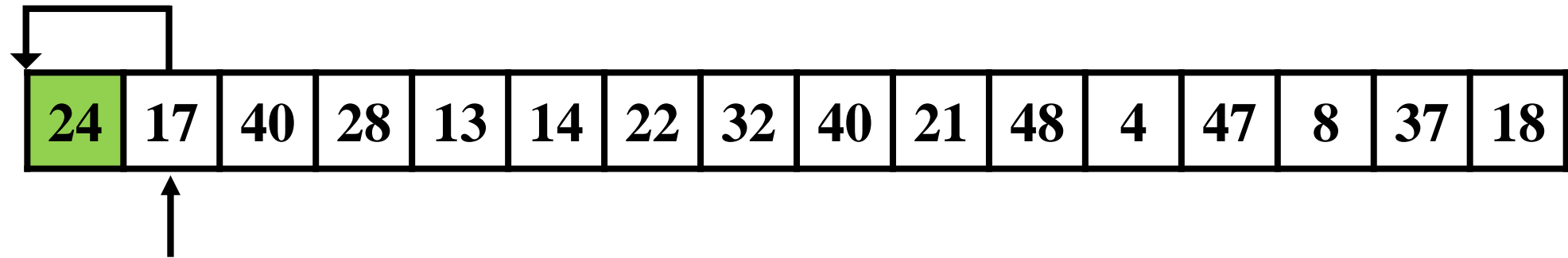
# 插入排序：算法实例

---



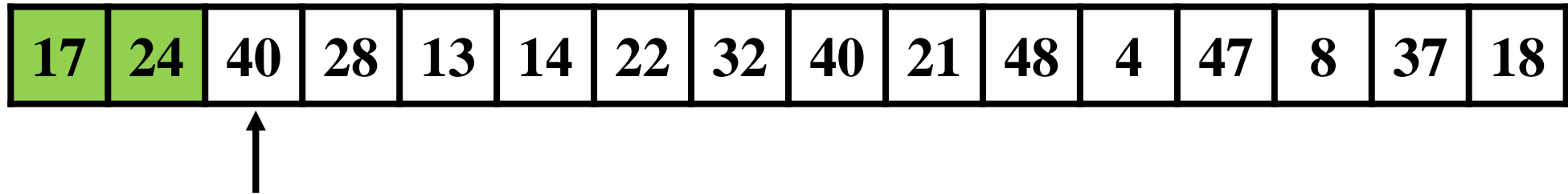
# 插入排序：算法实例

---



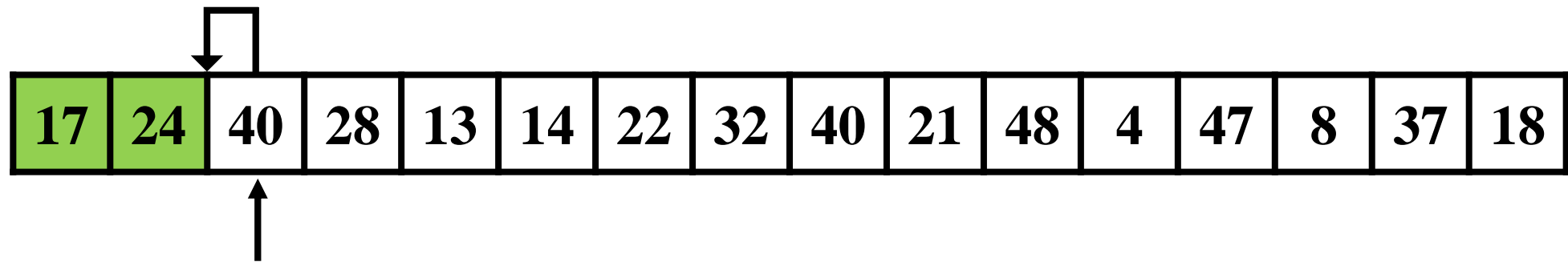
# 插入排序：算法实例

---



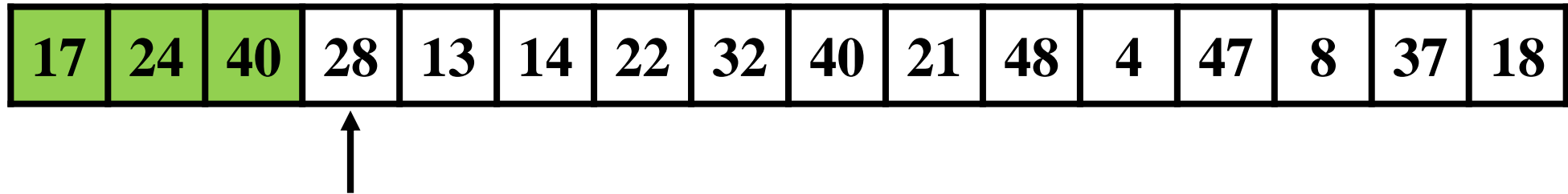
# 插入排序：算法实例

---



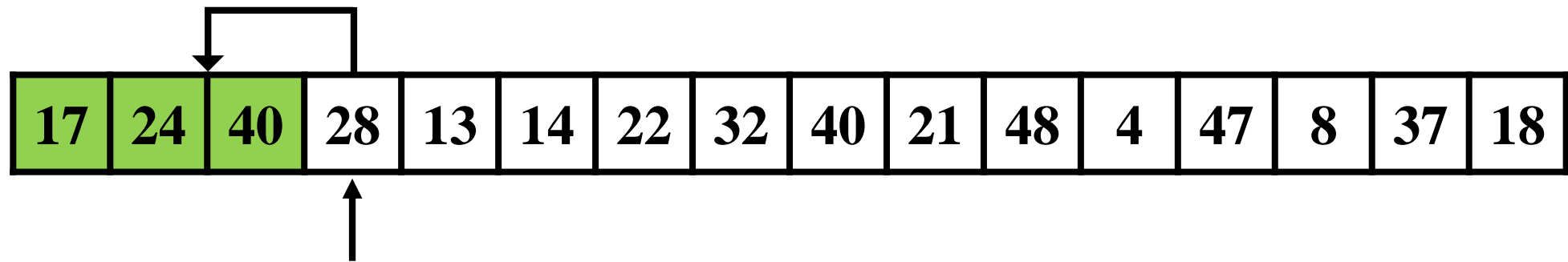
# 插入排序：算法实例

---



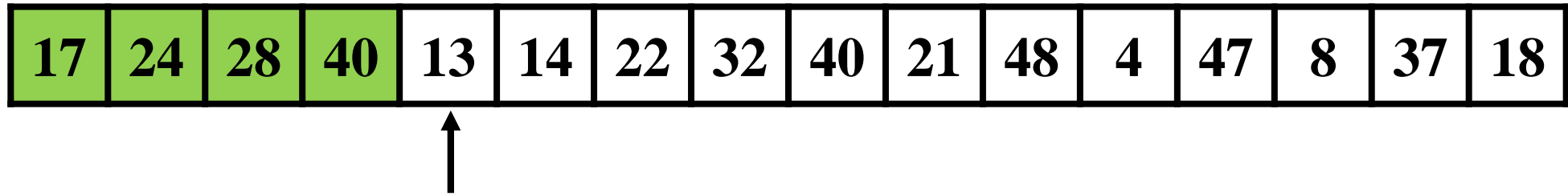
# 插入排序：算法实例

---



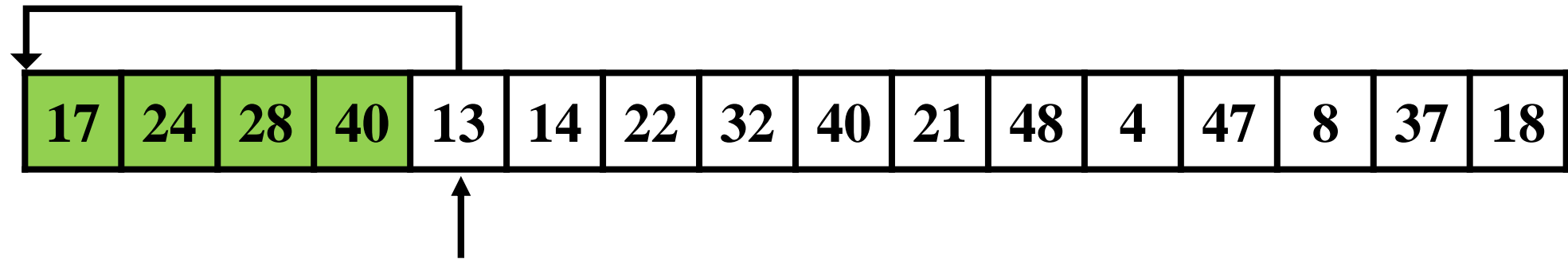
# 插入排序：算法实例

---



# 插入排序：算法实例

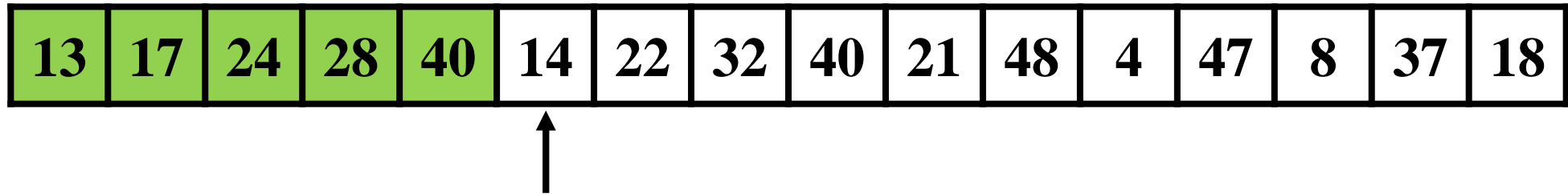
---





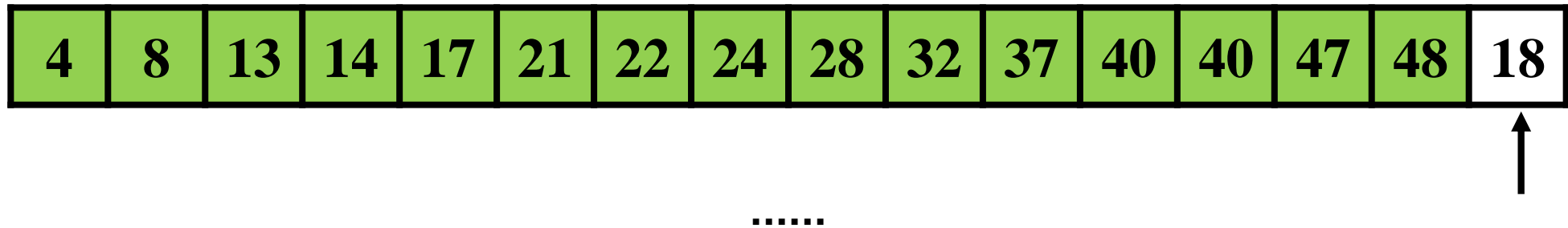
# 插入排序：算法实例

---



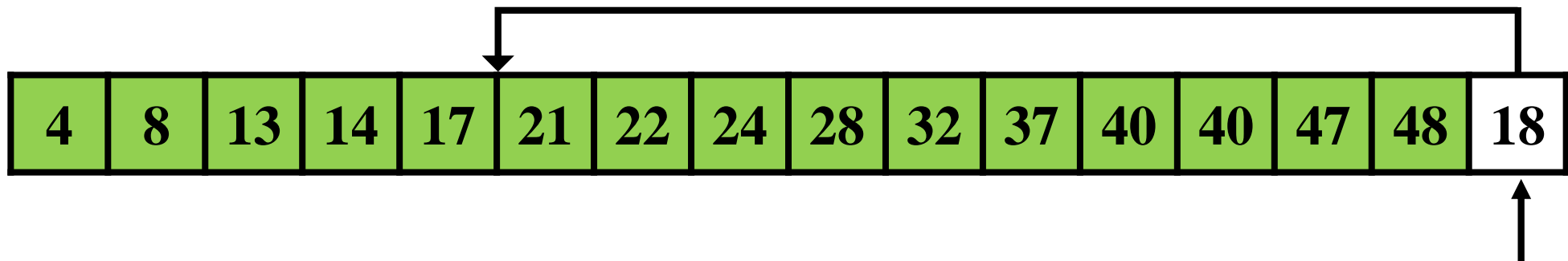
# 插入排序：算法实例

---



# 插入排序：算法实例

---



# 插入排序：算法实例

---

- 输入：

- $\langle 24, 17, 40, 28, 13, 14, 22, 32, 40, 21, 48, 4, 47, 8, 37, 18 \rangle$

- 输出：

- $\langle 4, 8, 13, 14, 17, 18, 21, 22, 24, 28, 32, 37, 40, 40, 47, 48 \rangle$

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

# 插入排序：算法实例

---

- 输入：

- $\langle 24, 17, 40, 28, 13, 14, 22, 32, 40, 21, 48, 4, 47, 8, 37, 18 \rangle$

- 输出：

- $\langle 4, 8, 13, 14, 17, 18, 21, 22, 24, 28, 32, 37, 40, 40, 47, 48 \rangle$

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

问题：排序问题是否存在其他算法？

# 选择排序

## 排序问题

### Sorting Problem

输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

## ● 选择排序算法

- 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

# 选择排序： 算法实例

---


- 输入： <24, 17, 40, 28, 13, 14, 22, 32, 40, 21, 48, 4, 47, 8, 37, 18>

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

# 选择排序：算法实例

---

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





# 选择排序：算法实例

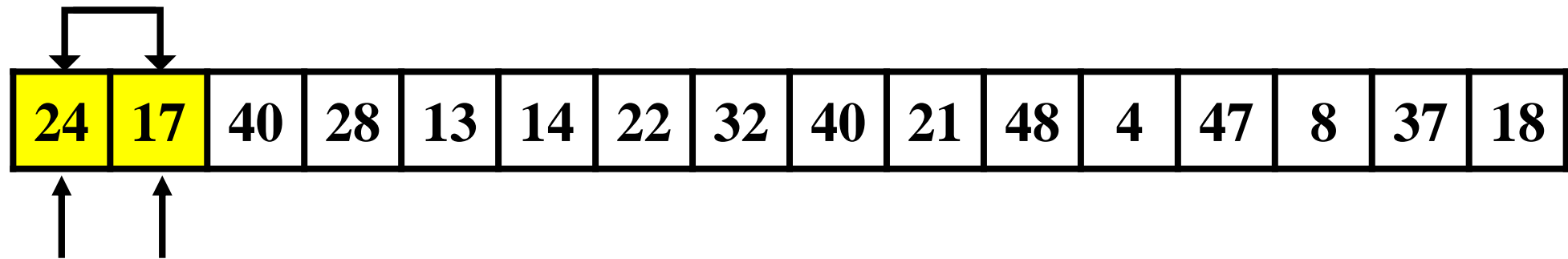
---

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

↑    ↑

# 选择排序：算法实例

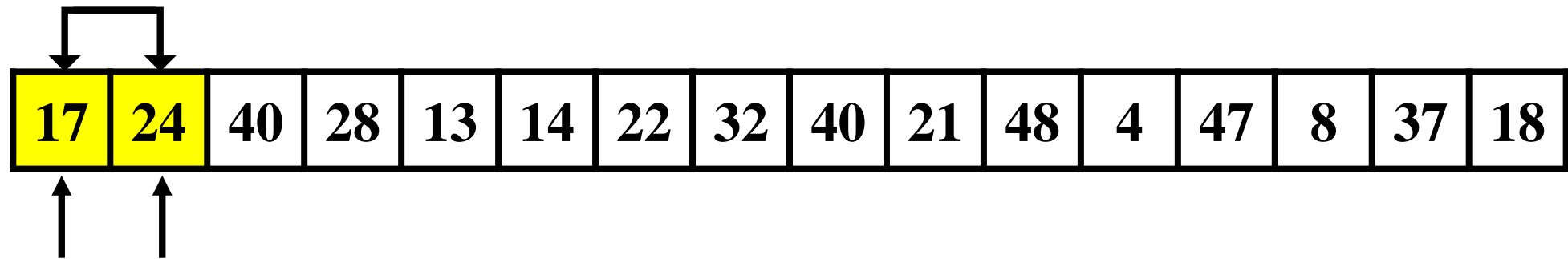
---



$24 > 17$

# 选择排序：算法实例

---



$17 < 24$

# 选择排序：算法实例

---

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



**$17 < 40$**

# 选择排序：算法实例

---

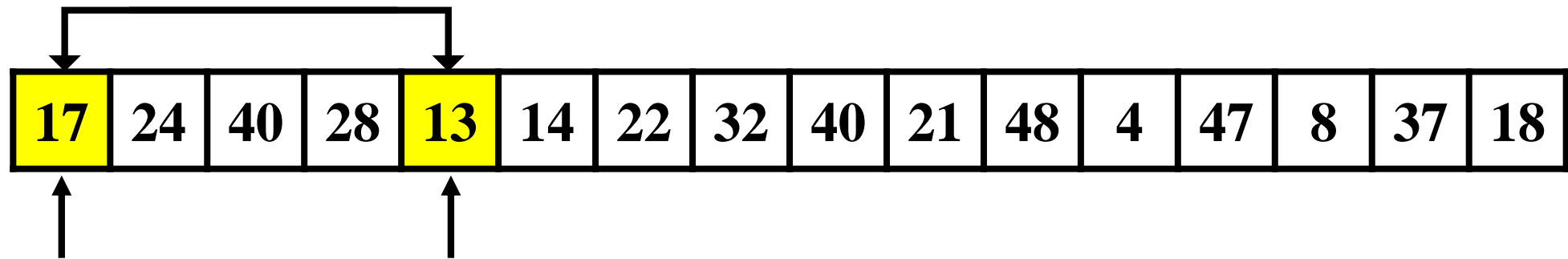
17	24	40	28	13	14	22	32	40	21	48	4	47	8	37	18
----	----	----	----	----	----	----	----	----	----	----	---	----	---	----	----



$17 < 28$

# 选择排序：算法实例

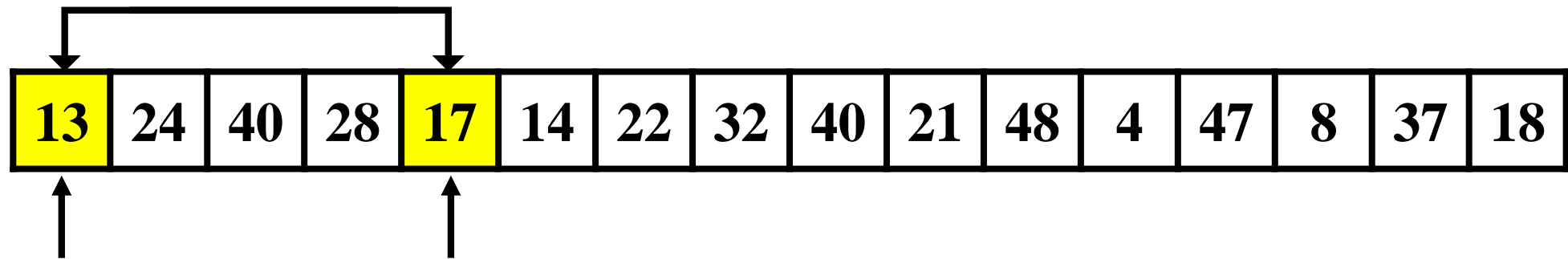
---



$17 > 13$

# 选择排序：算法实例

---



$13 < 17$

# 选择排序：算法实例

---

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

↑                      ↑

$13 < 14$



# 选择排序： 算法实例

---

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

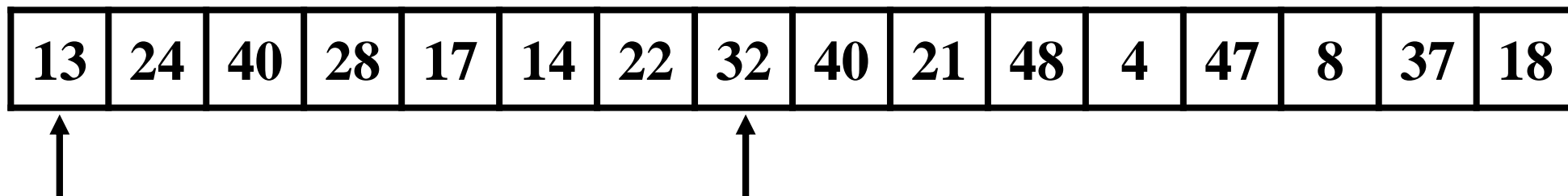
↑                      ↑

$13 < 22$

# 选择排序：算法实例

---

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



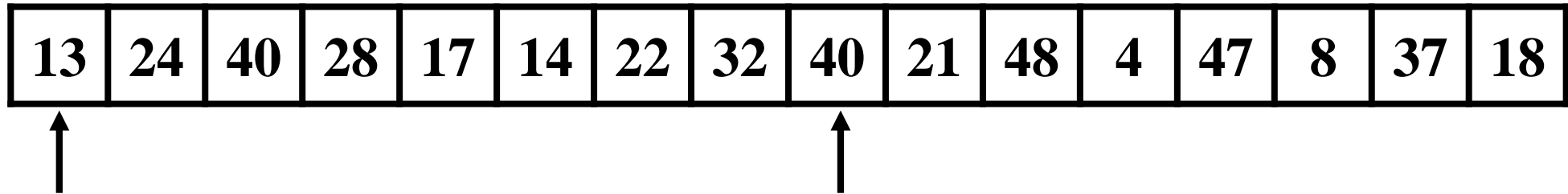
The diagram shows a horizontal array of 16 numbers. Two vertical arrows point upwards to the first and eighth cells of the array. Below the first arrow, the text '13 < 32' is displayed, indicating a comparison between the values in the first and eighth positions.

$13 < 32$

# 选择排序：算法实例

---

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



$13 < 40$

# 选择排序：算法实例

---

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



$13 < 21$

# 选择排序：算法实例

---

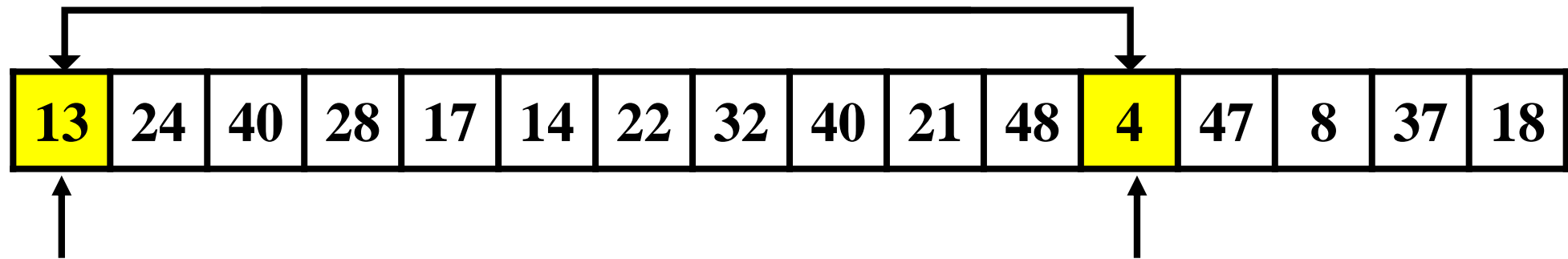
13	24	40	28	17	14	22	32	40	21	48	4	47	8	37	18
----	----	----	----	----	----	----	----	----	----	----	---	----	---	----	----



$13 < 48$

# 选择排序：算法实例

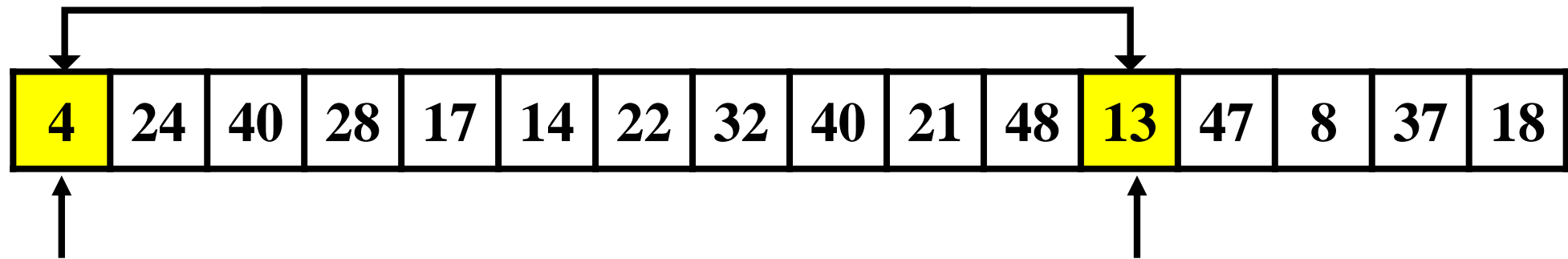
---



$13 > 4$

# 选择排序：算法实例

---



$4 < 13$

# 选择排序：算法实例

---

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



$4 < 47$



## 选择排序： 算法实例

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

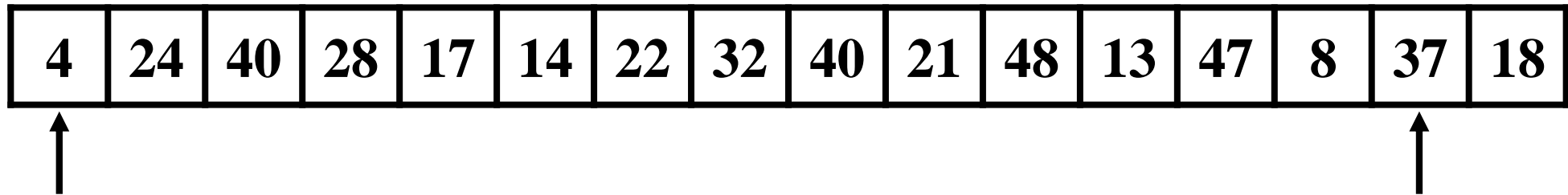


$$4 < 8$$

# 选择排序：算法实例

---

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



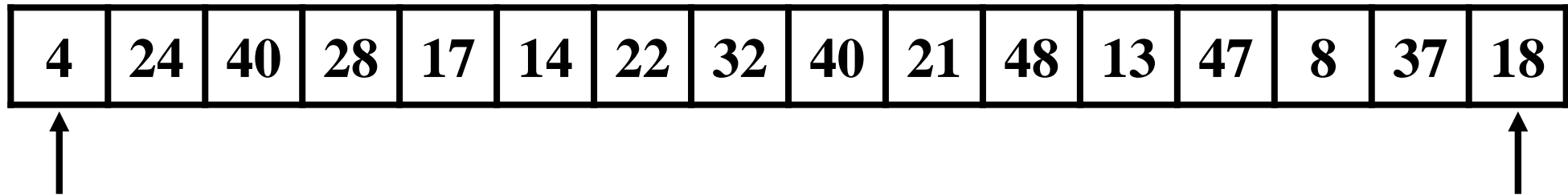
The diagram shows a horizontal array of 16 boxes, each containing a number. Below the first box (containing 4) and the 15th box (containing 37), there are upward-pointing arrows. Below these arrows, the text  $4 < 37$  is displayed.

$4 < 37$

# 选择排序：算法实例

---

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



The diagram shows a horizontal array of 16 boxes, each containing a number. The first box contains '4' and the last box contains '18'. Below the first box is an upward-pointing arrow, and below the last box is another upward-pointing arrow.

$4 < 18$

# 选择排序： 算法实例

---

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

# 选择排序：算法实例

---

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

# 选择排序：算法实例

---

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

# 选择排序： 算法实例

---

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

.....

# 选择排序： 算法实例

---

- 输入： <24, 17, 40, 28, 13, 14, 22, 32, 40, 21, 48, 4, 47, 8, 37, 18>
- 输出： <4, 8, 13, 14, 17, 18, 21, 22, 24, 28, 32, 37, 40, 40, 47, 48>

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



# 选择排序： 算法实例

---

- 输入： <24, 17, 40, 28, 13, 14, 22, 32, 40, 21, 48, 4, 47, 8, 37, 18>
- 输出： <4, 8, 13, 14, 17, 18, 21, 22, 24, 28, 32, 37, 40, 40, 47, 48>
- 满足
  - $4 \leq 8 \leq 13 \leq 14 \leq 17 \leq 18 \leq 21 \leq 22 \leq 24 \leq 28 \leq 32 \leq 37 \leq 40 \leq 40 \leq 47 \leq 48$

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

# 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

# 算法的性质

---



有穷性



确定性



可行性

# 算法的性质

有穷性

确定性

可行性

- 含义

- 算法必须在有限个计算步骤后终止

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义
  - 算法必须在有限个计算步骤后终止
- 反例：
  - 给定输入数组，不断交换首尾元素的位置

## 排序问题

### Sorting Problem

输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义

- 算法必须在有限个计算步骤后终止

- 反例：

- 给定输入数组，**不断交换**首尾元素的位置

动作序列没有终结

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义

- 算法必须在有限个计算步骤后终止

- 反例：

- 给定输入数组，不断交换首尾元素的位置

- 正例：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

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义

- 算法必须在有限个计算步骤后终止

- 反例：

- 给定输入数组，不断交换首尾元素的位置

- 正例：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

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

遍历次数至多与数组元素个数相同



# 算法的性质

有穷性

确定性

可行性

- 含义
  - 算法必须是没有歧义的

## 排序问题

### Sorting Problem

输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义
  - 算法必须是没有歧义的
- 反例：
  - 对于给定输入数组，交换两个数的位置

## 排序问题

### Sorting Problem

输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义
  - 算法必须是没有歧义的
- 反例：
  - 对于给定输入数组，交换两个数的位置

没有具体指明是哪两个数

## 排序问题

### Sorting Problem

输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义

- 算法必须是没有歧义的

- 反例：

- 对于给定输入数组，交换两个数的位置

- 正例：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

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义

- 算法必须是没有歧义的

- 反例：

- 对于给定输入数组，交换两个数的位置

- 正例：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

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

每一个步骤都是确定的

# 算法的性质

有穷性

确定性

可行性

- 含义

- 可以机械地一步一步执行基本操作步骤

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义
  - 可以机械地一步一步执行基本操作步骤
- 反例：
  - 将大元素放数组后部，小元素放数组前部

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义

- 可以机械地一步一步执行基本操作步骤

- 反例：

- 将大元素放数组后部，小元素放数组前部

描述含糊，不可拆解为基本操作步骤

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$



# 算法的性质

有穷性

确定性

可行性

- 含义

- 可以机械地一步一步执行基本操作步骤

- 反例：

- 将大元素放数组后部，小元素放数组前部

- 正例：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

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

# 算法的性质

有穷性

确定性

可行性

- 含义
  - 可以机械地一步一步执行基本操作步骤
- 反例：
  - 将大元素放数组后部，小元素放数组前部
- 正例：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

## 排序问题

### Sorting Problem

#### 输入

- 包含 $n$ 个数字的数组  $\langle a_1, \dots, a_n \rangle$

#### 输出

- 升序排列的数组

$$\langle a'_1, a'_2, \dots, a'_n \rangle$$

$$\text{满足 } a'_1 \leq a'_2 \leq \dots \leq a'_n$$

算法可一步步地执行完成