

算法的由来

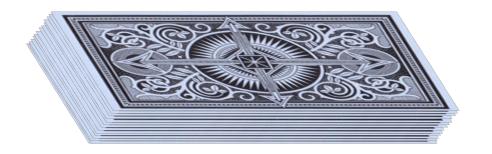
算法的定义

算法的性质

算法的表示

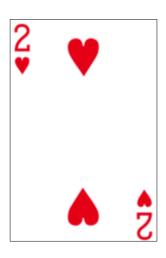
算法的分析





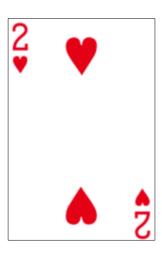


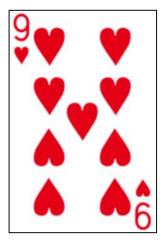






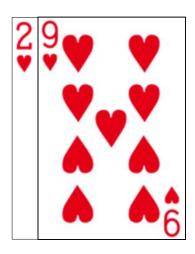






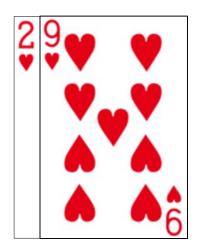


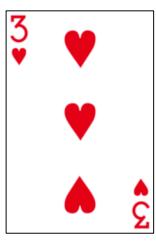






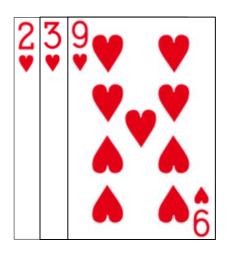






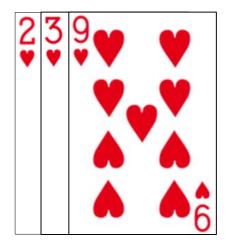


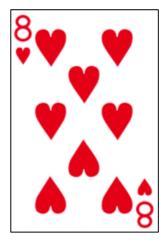






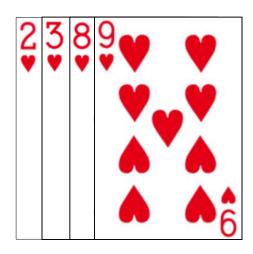






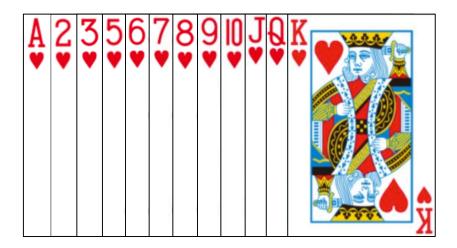




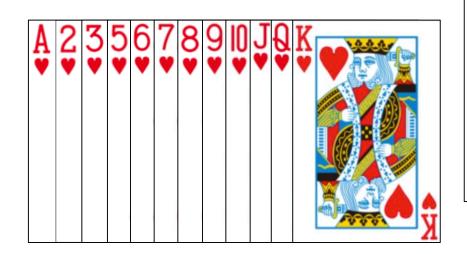






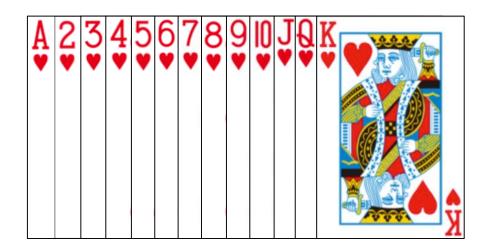














- 定义
  - 给定数据输入,计算满足某种性质输出的问题



- 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  $< a_1, ..., a_n >$ 

### 输出

• 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



- 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

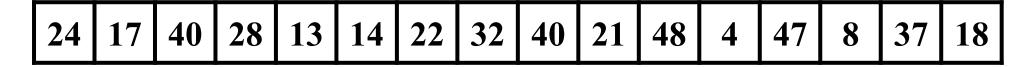
• 包含n个数字的数组  $< a_1, ..., a_n >$ 

### 输出

• 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 

输人: 包含16个数字的数组





- 定义
  - 给定数据输入,计算满足某种性质输出的问题
- 示例

# 一种诗间题是一种计算问题。

**Sorting Problem** 

#### 输入

包含n个数字的数组  $< a_1, ..., a_n >$ 

### 输出

升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 

输入: 包含16个数字的数组 **17 40 28** 13 48 **37** 18 **22 32 40 14** 21

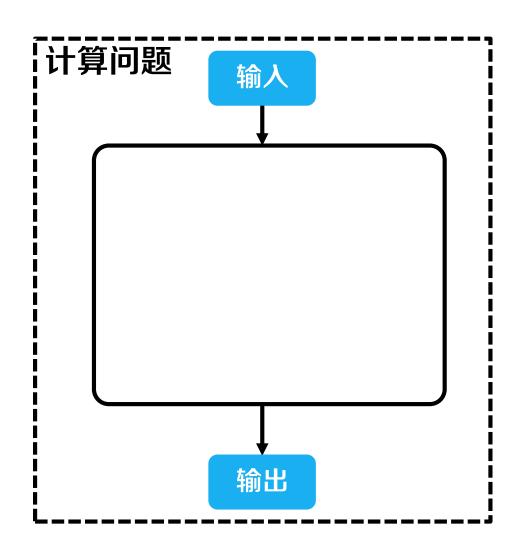
输出: 满足升序性质的输入数组

13

32

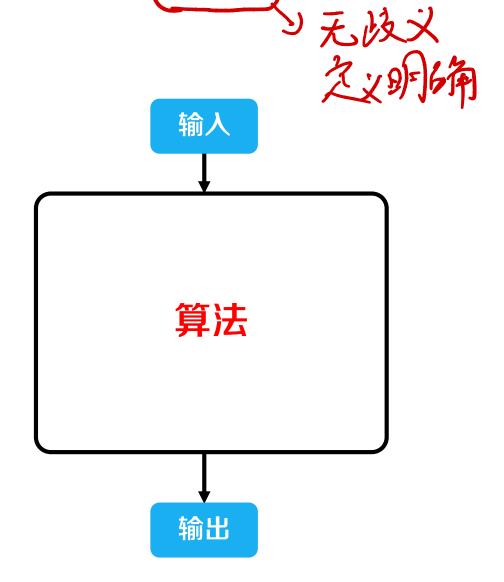


• 给定计算问题



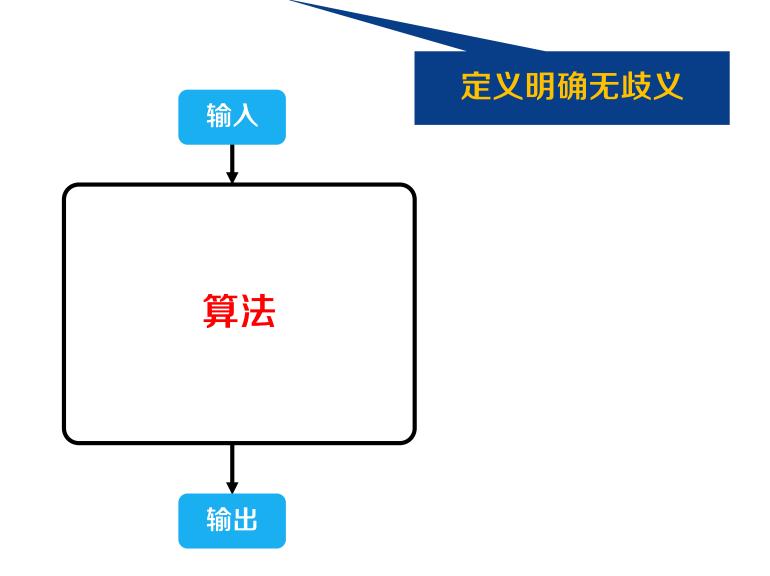


• 给定计算问题,算法是一系列良定义的计算步骤



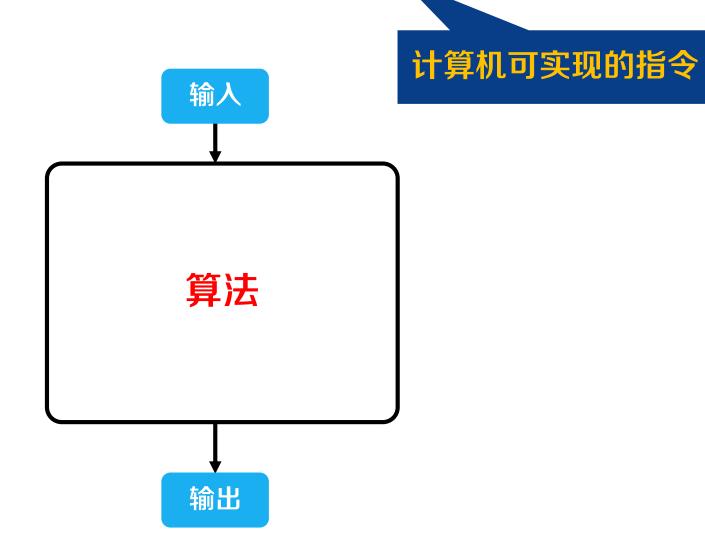


• 给定计算问题,算法是一系列良定义的计算步骤



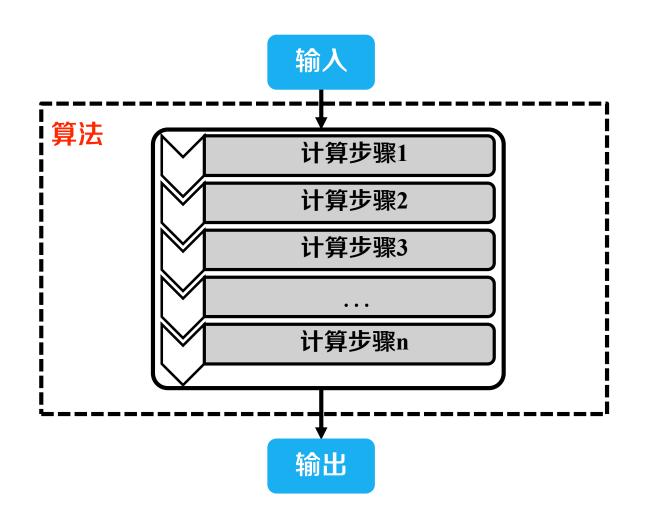


• 给定计算问题,算法是一系列良定义的计算步骤





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  $< a_1, ..., a_n >$ 

### 输出

• 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 

- 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



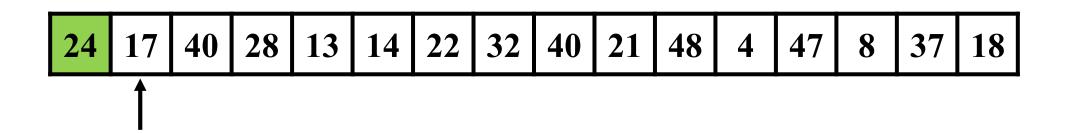
• 输入:

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

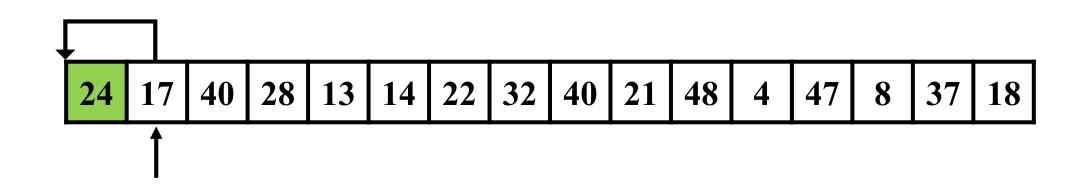




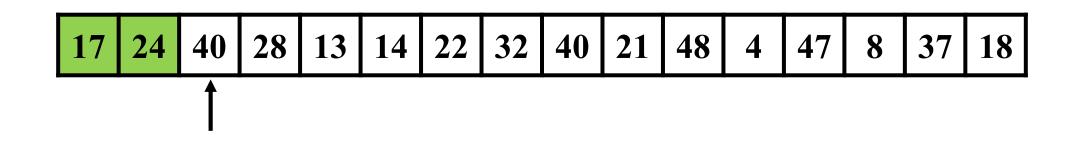




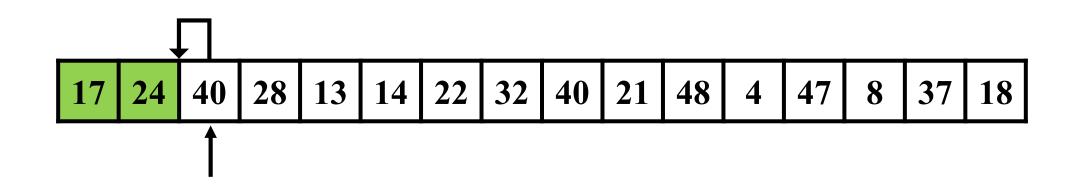




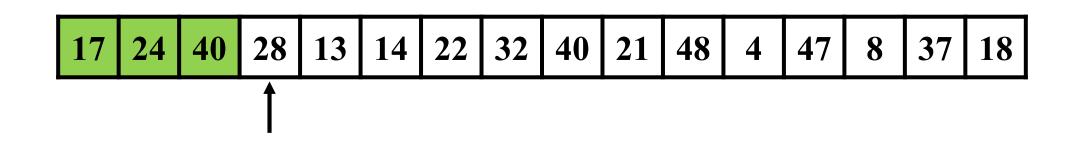




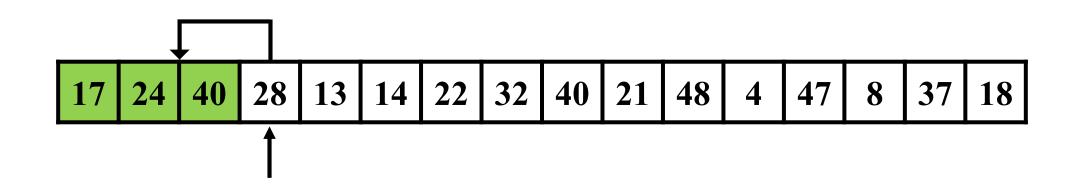




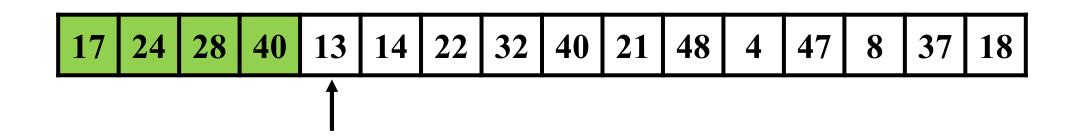




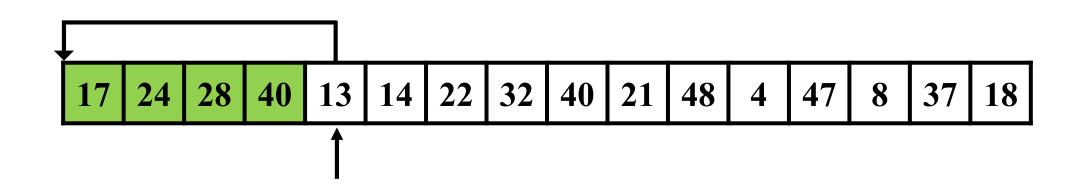




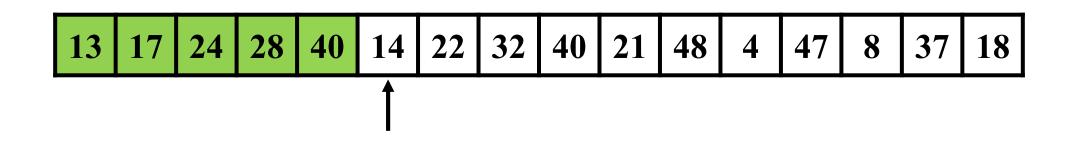




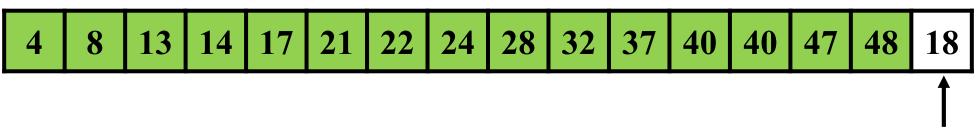






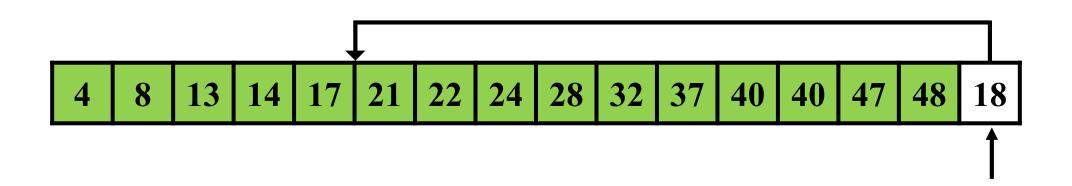






. . . . . .







• 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

# 插入排序: 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

问题:排序问题是否存在其他算法?



## 排序问题

### **Sorting Problem**

### 输入

• 包含n个数字的数组  $< a_1, ..., a_n >$ 

## 输出

• 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 

# • 选择排序算法

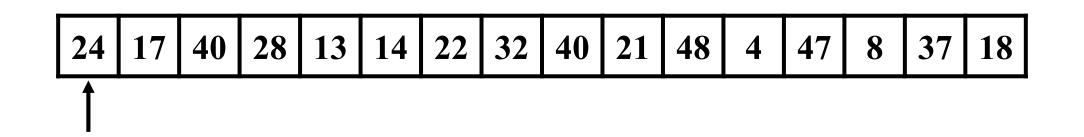
- 第一次遍历找到最小元素
- 第二次在剩余数组中遍历找到次小元素
- ---
- 第*n*次在剩余数组中遍历找到第*n*小元素



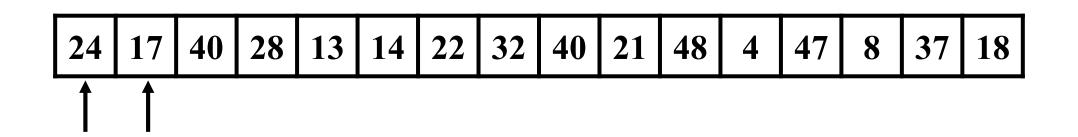
• 输入: <24, 17, 40, 28, 13, 14, 22, 32, 40, 21, 48, 4, 47, 8, 37, 18>

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

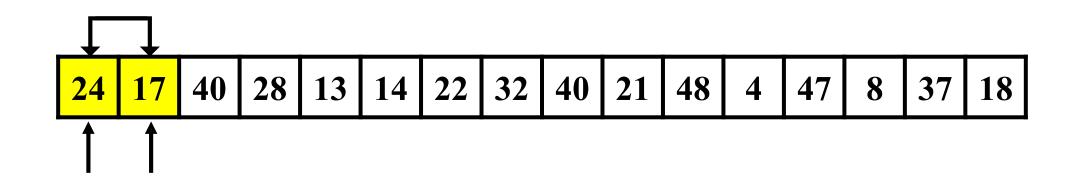






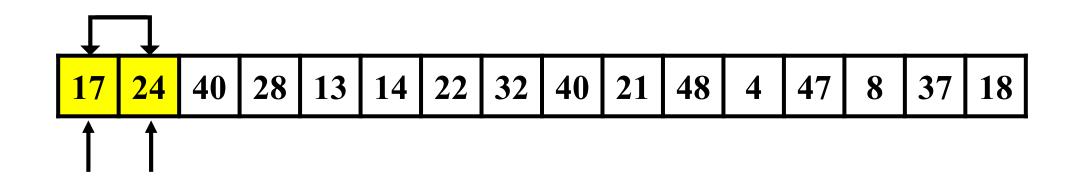




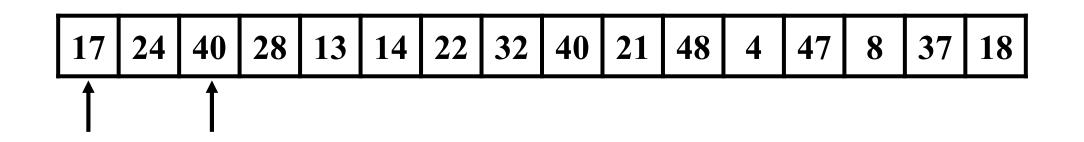


24 > 17

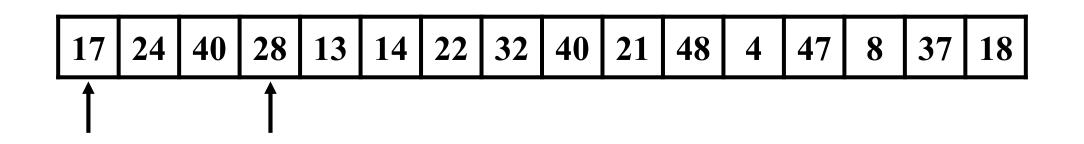




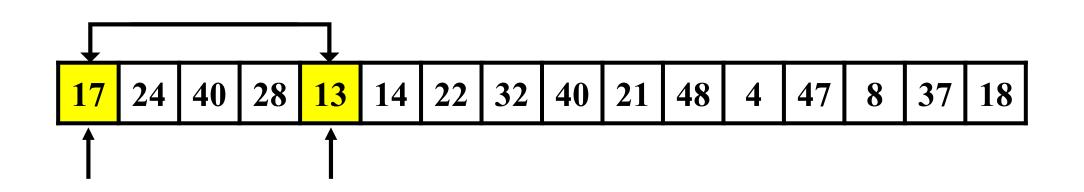






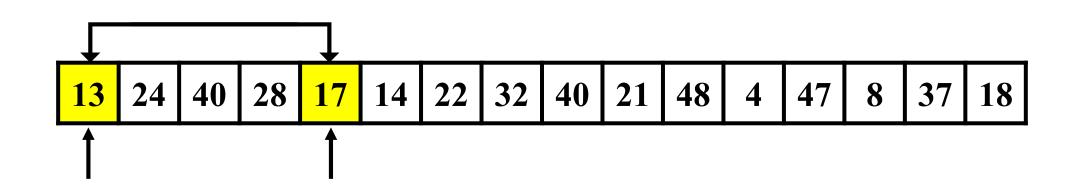




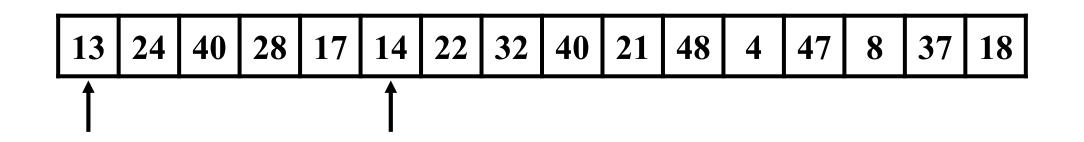


17 > 13

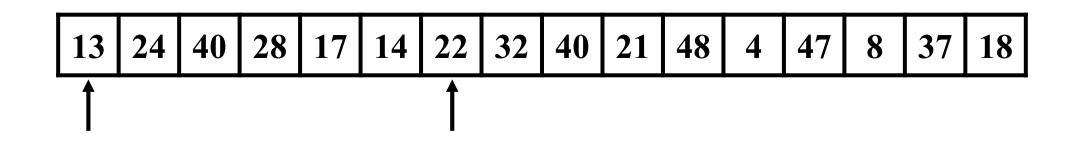




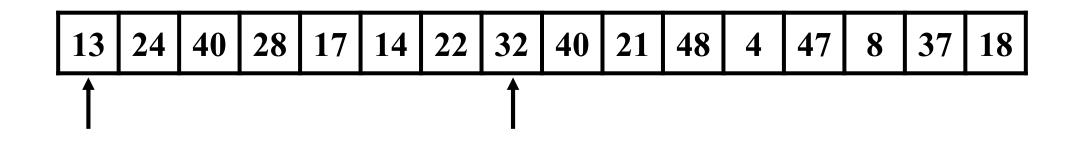




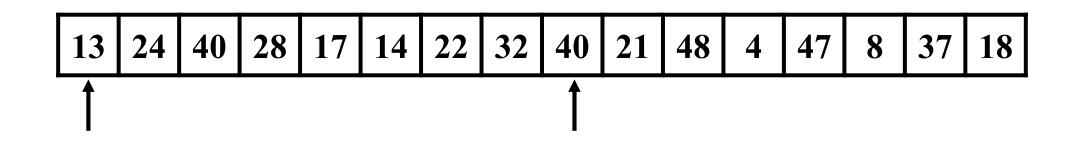




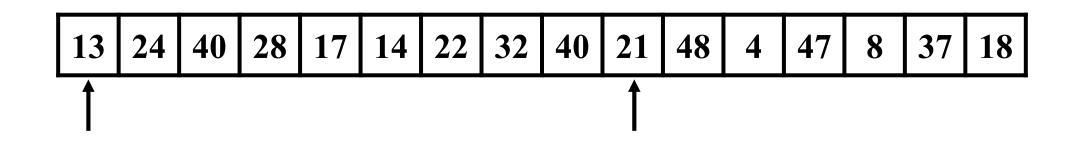




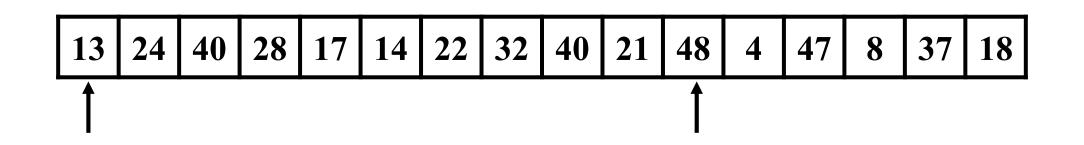




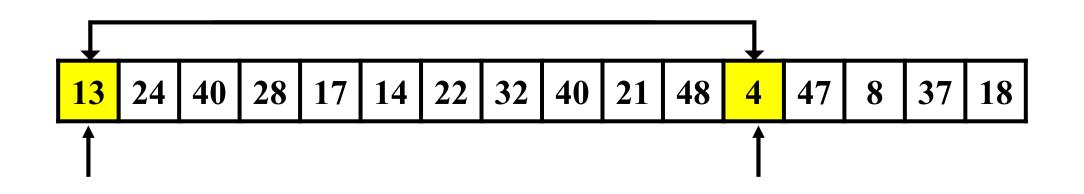






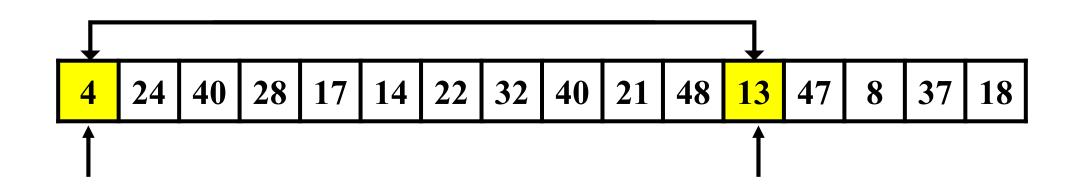




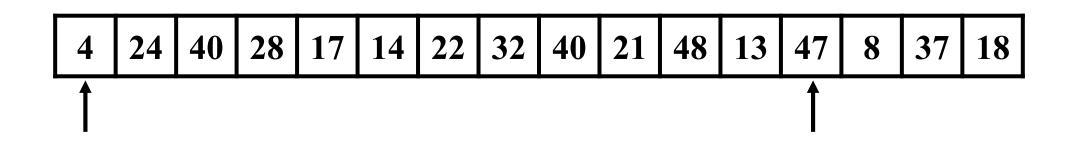


13 > 4

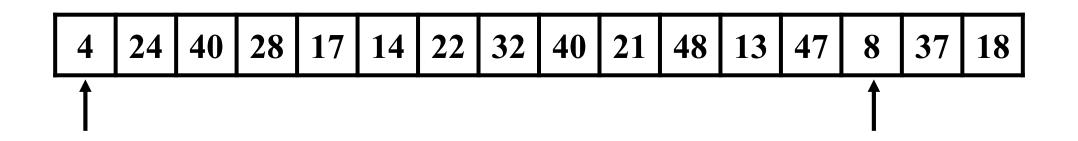




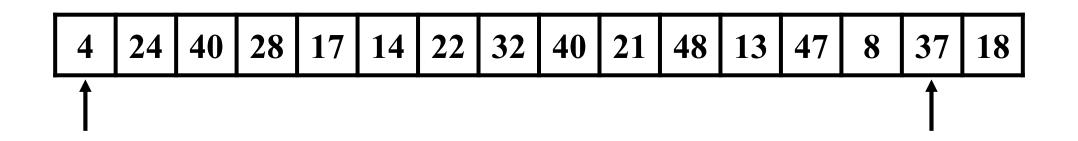




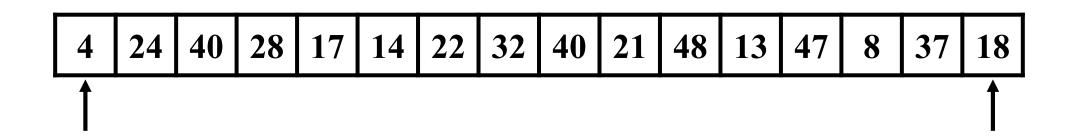








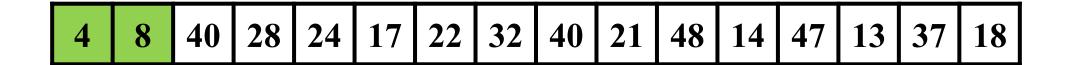






 4
 24
 40
 28
 17
 14
 22
 32
 40
 21
 48
 13
 47
 8
 37
 18







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



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

• • • • •



• 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>





• 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

•  $4 \le 8 \le 13 \le 14 \le 17 \le 18 \le 21 \le 22 \le 24 \le 28 \le 32 \le 37 \le 40 \le 40 \le 47 \le 48$ 

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



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 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**

- 包含n个数字的数组  $< a_1, ..., a_n >$  输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



### 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**

- 包含n个数字的数组  $< a_1, ..., a_n >$  输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



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  $< a_1, ..., a_n >$  输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



#### 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**

- 包含n个数字的数组  $< a_1, ..., a_n >$ 输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



#### 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  $< a_1, ..., a_n >$ 输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 

遍历次数至多与数组元素个数相同



确定性

可行性

- 含义
  - 算法必须是没有歧义的

# 排序问题

# **Sorting Problem**

- 包含n个数字的数组  $< a_1, ..., a_n >$  输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



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

- 包含n个数字的数组  $< a_1, ..., a_n >$  输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



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**

- 包含n个数字的数组  $< a_1, ..., a_n >$  输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



#### 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**

- 包含n个数字的数组  $< a_1, ..., a_n >$ 输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



#### 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  $< a_1, ..., a_n >$ 输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 

### 每一个步骤都是确定的



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**

- 包含n个数字的数组  $< a_1, ..., a_n >$ 输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



### 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**

- 包含n个数字的数组  $< a_1, ..., a_n >$  输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



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

- 包含n个数字的数组  $< a_1, ..., a_n >$  输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



### 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**

- 包含n个数字的数组  $< a_1, ..., a_n >$ 输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 



#### 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  $< a_1, ..., a_n >$ 输出
- 升序排列的数组

$$< a_1', a_2', ..., a_n' >$$
  
满足 $a_1' \le a_2' \le \cdots \le a_n'$ 

算法可一步步地执行完成