

# 数据结构与算法实战

## 第二讲 数组与字符串

### 2-1 数组

# 2.1.1 数组的定义与使用

- 所谓“数组”：线性表的一种实现
- 各种语言都有“数组”
- 怎样定义、使用？ C / C++ / Java / Python

# 2.1.1 数组的定义与使用

## C语言

- `int array[N]` 或 `int *array = malloc()`
- “原生态”的数组，与“指针”关系密切
- 数组名 不是 指针 ！
- 如何传递参数？

# 2.1.1 数组的定义与使用

C++

- `int array[N]` 或 `int *array = new int[N]`
- 还可以使用STL: `vector`

# 2.1.1 数组的定义与使用

Java

- `int [] array = new int [N]`
- 还可以使用容器类: `ArrayList`

# 2.1.1 数组的定义与使用

Python

- “以上三个好麻烦……”
- 用序列（列表）搞定~ `array = []`

## 2.1.2 数组元素的查找

- 常见操作：查找、插入、删除
- “查找”包括：
  - 找x的序号（是否包含x）
  - 找第k个元素
  - 找max / min
  - 统计个数

# 2.1.3 数组实例

高精度加法：

求两个100位的十进制数字之和



## 2.1.3 数组实例

高精度加法：

求两个100位的十进制数字之和

C:

我模拟竖式加法！

# 2.1.3 数组实例

高精度加法：

求两个100位的十进制数字之和

Java：

我有 BigInteger ！

# 数据结构与算法实战

## 第二讲 数组与字符串

### 2-2 二维数组

## 2.2.1 二维数组的定义与使用

- 二维数组的实质：数组的数组
- 逻辑上可看作二维，其实并不是“二维”
- 怎样定义、使用？ C / C++ / Java / Python

# 2.2.1 二维数组的定义与使用

## C语言

- `int array[][]` 与 `int **array` : `array[i][j]` 都合法？
- 如何传递参数

# 2.2.1 二维数组的定义与使用

C语言 `int array[][]` 与 `int **`

# 2.2.1 二维数组的定义与使用

Java

- 如何创建二维数组?
- 长度可不同

## 2.2.2 二维数组实例： 大炮打蚊子

现在，我们用大炮来打蚊子：蚊子分布在一个 $M \times N$ 格的二维平面上，每只蚊子占据一格。向该平面的任意位置发射炮弹，炮弹的杀伤范围如下示意：

```
  o
oxo
  o
```

其中，**x** 为炮弹落点中心，**o** 为紧靠中心的四个有杀伤力的格子范围。若蚊子被炮弹命中（位于**x**格），一击毙命，若仅被杀伤（位于**o**格），则损失一半的生命力。也就是说，一次命中或者两次杀伤均可消灭蚊子。现在给出蚊子的分布情况以及连续**k**发炮弹的落点，给出每炮消灭的蚊子数。

### 输入格式：

第一行为两个不超过20的正整数**M**和**N**，中间空一格，表示二维平面有**M**行、**N**列。

接下来**M**行，每行有**N**个**0**或者**#**字符，其中**#**表示所在格子有蚊子。

接下来一行，包含一个不超过400的正整数**k**，表示发射炮弹的数量。

最后**k**行，每行包括一发炮弹的整数坐标**x**和**y**（ $0 \leq x < M$ ， $0 \leq y < N$ ），之间用一个空格间隔。

### 输出格式：

对应输入的**k**发炮弹，输出共有**k**行，第**i**行即第**i**发炮弹消灭的蚊子数。



## 2.2.2 二维数组实例：大炮打蚊子


# 数据结构与算法实战

## 第二讲 数组与字符串

2-3 字符串

## 2.3.1 字符串的定义与使用

- “字符串”：一串字符
- 各种语言都有“字符串”
- 怎样定义、使用？ C / C++ / Java / Python

# 2.3.1 字符串的定义与使用

## C语言

- "abcdefg" 或 字符数组
- 字符串 和 字符数组 不同
- 注意安全性

# 2.3.1 字符串的定义与使用

C++

- "abcdefg" 或 字符数组
- string

# 2.3.1 字符串的定义与使用

Java

- 字符数组（比较少用）
- String

# 2.3.1 字符串的定义与使用

Python

- “我的字符串简单易用”

## 2.3.2 字符串的常用操作

- 比较、拼接、复制
- 子串：截取、查找（定位）、替换、计数
- 匹配



## 2.3.3 正则表达式与串匹配

- 正则表达式
- 自动机
- 字符串匹配

## 2.3.3 正则表达式与串匹配

### 正则表达式

- $a$     $a|b$     $[abc]$     $a-z$
- $a^+$     $[a-z0-9]$
- $[a-z0-9]^+$

## 2.3.3 正则表达式与串匹配

- 自动机
- 非确定型有限状态自动机 (Nondeterministic Finite Automata, NFA)
- 确定型有限状态自动机 (Deterministic Finite Automata, DFA)
- 图灵机?

## 2.3.3 正则表达式与串匹配

字符串匹配

- 写一个自动机
- 利用正则表达式