

# 数据结构与算法实战

## 第六讲 查找与散列

### 6.1 常用查找方法



其实，散列属于查找

# 6.1.1 顺序查找

适用于无序的顺序表（或链表）

注意区分两个词：ordered(或sorted) V.S. sequenced


0	1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	x	i	j

- 表中找值为x的元素的下标/地址/引用
- 判断元素x是否在表中

# 6.1.1 顺序查找

适用于无序的顺序表（或链表）

注意区分两个词：ordered(或sorted) V.S. sequenced



0	1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	x	i	j

- 表中找值为x的元素的下标/地址/引用

$O(N)$

- 判断元素x是否在表中

# 6.1.2 二分查找（折半查找）

适用于有序的顺序表（不适用于链表！）

0	1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	x	i	j

- 表中找值为x的元素的下标/地址/引用
- 判断元素x是否在表中

## 6.1.2 二分查找（折半查找）

适用于有序的顺序表（不适用于链表！）

low		mid				high			
↓		↓				↓			
0	1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	x	i	j

- 表中找值为x的元素的下标/地址/引用
- 判断元素x是否在表中

$O(\log N)$

# 6.1.3 查找树

- ~~二叉搜索树 (BST)~~
- AVL树
- 红黑树
- B树、B+树
- .....

$O(\log_k N)$

# 数据结构与算法实战

## 第六讲 查找与散列

### 6.2 散列

# 6.2.1 散列的概念

先来解决几个小问题：

1、容量为100的教室里，应有60名学生上课，学生学号为1-60号。

如何能一眼看出学号为 $x$ 的学生是否到课？



# 6.2.1 散列的概念

先来解决几个小问题：

2、容量为100的教室里，有60名学生上课，假设学生身高范围在140cm ~ 220cm之间，数值均为整数且各不相同。

如何能一眼看出教室里是否有一名学生身高为 $x$ ？

# 6.2.1 散列的概念

目标和手段：

- 为了更快查找 ——  $O(1)$
- 拿空间换时间 —— 时间效率、空间效率
- 计算空间位置要快 ——  $O(1)$

散列：

- “键-值对” (key-value pair) 的集合。

# 6.2.1 散列的概念

散列：“键-值对” (key-value pair) 的集合

两个关键问题

- 散列函数 什么是好的散列函数？
- 解决冲突 能想出哪些办法？

## 6.2.2 冲突解决

- 开放定址 (Open Addressing)
- 分离链 (Separate Chaining)
- 其它 (公共溢出区等)

# 6.2.2 冲突解决

开放定址  $h(x) = x \bmod 11$

$x = \{12, 2, 17, 28, 6, 23\}$      $h(x) = \{1, 2, 6, 6, 6, 1\}$

0	1	2	3	4	5	6	7	8	9	10

线性探测、平方探测……

## 6.2.2 冲突解决

分离链  $h(x) = x \bmod 11$

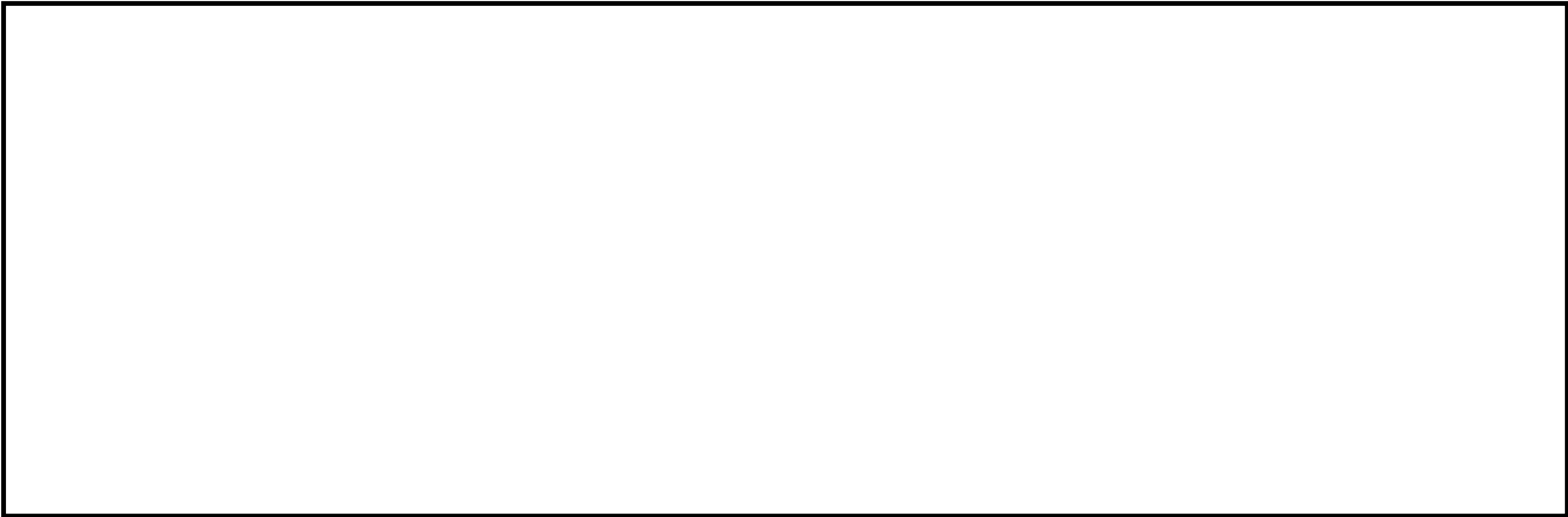
$$\mathbf{x} = \{12, 2, 17, 28, 6, 23\} \quad \mathbf{h}(\mathbf{x}) = \{1, 2, 6, 6, 6, 1\}$$
[illegible]

# 6.2.2 冲突解决

公共溢出区       $h(x) = x \bmod 11$

$x=\{12, 2, 17, 28, 6, 23\}$        $h(x)=\{1, 2, 6, 6, 6, 1\}$

0	1	2	3	4	5	6	7	8	9	10



## 6.2.3 各语言实现散列

- C
- C++
- Java
- Python