



# 数据结构与算法（十）

张铭 主讲

采用教材：张铭，王腾蛟，赵海燕 编写  
高等教育出版社，2008.6（“十一五”国家级规划教材）

<http://www.jpk.pku.edu.cn/pkujpk/course/sjjg>



## 10.1 线性表的检索

# 第十章 检索

- 10.1 线性表的检索
- 10.2 集合的检索
- 10.3 散列表的检索
- 总结



# 集合

- 集合 ( set ) : 由若干个确定的、相异的对象 ( element ) 构成的整体
- 集合的检索 : 确定一个值是不是某个集合的元素

## 10.2 集合的检索

	运算名称	数学运算 符号	计算机运算符号
算术 运算	并	$\cup$	$+$ 、 $ $ 、 $OR$
	交	$\cap$	$*$ 、 $\&$ 、 $AND$
	差	$-$	$-$
	相等	$=$	$==$
	不等	$\neq$	$!=$
逻辑 运算	包含于	$\subseteq$	$\leq$
	包含	$\supseteq$	$\geq$
	真包含于	$\subset$	$<$
	真包含	$\supset$	$>$
	属于	$\in$	$IN$ 、 $at$



# 集合的抽象数据类型

```
template<size_t N>                                // N为集合的全集元素个数
class mySet {
public:
    mySet();                                       // 构造函数
    mySet(ulong X);
    mySet<N>& set();                               // 设置元素属性
    mySet<N>& set(size_t P, bool X = true);
    mySet<N>& reset();                             // 把集合设置为空
    mySet<N>& reset(size_t P);                     // 删除元素P
    bool at(size_t P) const;                       // 属于运算
    size_t count() const;                          // 集合中元素个数
    bool none() const;                            // 判断是否空集
```



# 集合的抽象数据类型

```
bool operator==(const mySet<N>& R) const;           // 等于
bool operator!=(const mySet<N>& R) const;           // 不等
bool operator<=(const mySet<N>& R) const;           // 包含于
bool operator< (const mySet<N>& R) const;           // 真包含于
bool operator>=(const mySet<N>& R) const;           // 包含
bool operator> (const mySet<N>& R) const;           // 真包含

friend mySet<N> operator&(const mySet<N>& L, const mySet<N>& R); // 交
friend mySet<N> operator|(const mySet<N>& L, const mySet<N>& R); // 并
friend mySet<N> operator-(const mySet<N>& L, const mySet<N>& R); // 差
friend mySet<N> operator^(const mySet<N>& L, const mySet<N>& R); // 异或
};
```

# 集合的检索

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0

- 用位向量来表示集合
  - 对于密集型集合（数据范围小，而集合中有效元素个数比较多）

# 例：计算0到15之间所有的奇素数

奇数：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

素数：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0

奇素数：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	0

&amp;

||





## 例：集合的无符号数表示

- 全集元素个数  $N=40$  的集合
- 集合 $\{35, 9, 7, 5, 3, 1\}$  用 2 个 ulong 表示

```
0000 0000 0000 0000 0000 0000 0000 1000
0000 0000 0000 0000 0000 0010 1010 1010
```

不够一个 ulong , 左补 0



```
typedef unsigned long ulong;
enum {
    // unsigned long数据类型的位的数目
    NB = 8 * sizeof (ulong),
    // 数组最后一个元素的下标
    LI = N == 0 ? 0 : (N - 1) / NB
};
// 存放位向量的数组
ulong A[LI + 1];
```

## 设置集合元素

```
template<size_t N>
mySet<N>& mySet<N>::set(size_t P, bool X) {
    if (X)                // X为真，位向量中相应值设为1
        A[P / NB] |= (ulong)1 << (P % NB);
                        // P对应的元素进行按位或运算
    else    A[P / NB] &= ~((ulong)1 << (P % NB));
                        // X为假，位向量中相应值设为0
    return (*this);
}
```



## 集合的交运算 "&"

```
template<size_t N>
mySet<N>& mySet<N>::operator&=(const mySet<N>& R)
{ // 赋值交
    for (int i = LI; i >= 0; i--) // 从低位到高位
        A[i] &= R.A[i];         // 以ulong元素为单位按位交
    return (*this);
}

template<size_t N>
mySet<N> operator&(const mySet<N>& L, const mySet<N>& R)
{ //交
    return (mySet<N>(L) &= R);
}
```



## 思考

- 集合还可以用哪些技术来实现？
- 调研 STL 中集合的各种实现方法。



# 数据结构与算法

谢谢聆听

国家精品课“数据结构与算法”

<http://www.jpk.pku.edu.cn/pkujpk/course/sjjg/>

张铭，王腾蛟，赵海燕

高等教育出版社，2008. 6。“十一五”国家级规划教材