

9. 词典

(xb1) 位图：数据结构

邓俊辉

deng@tsinghua.edu.cn

集合及其接口

❖ 集合，也可视作一种抽象数据类型

❖ 这里不妨考查其中的特例——整数集

❖ 依散列的思想观之，所有离散集

或者显式地本身就是整数集

或者隐式地可转换为整数集

❖ ADT `void set(int k);` //将整数k加入当前集合

`void clear(int k);` //从当前集合中排除整数k

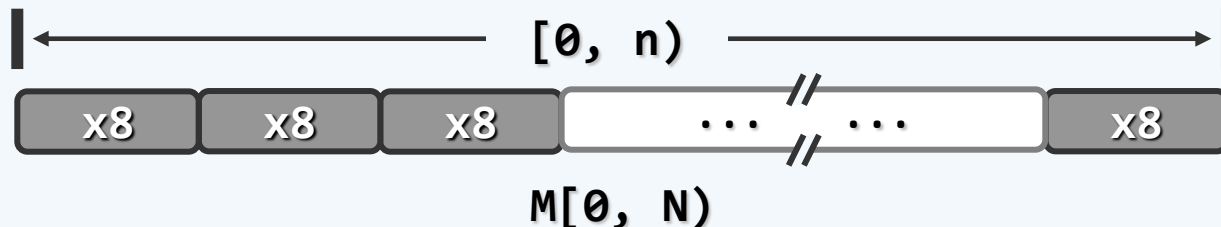
`bool test(int k);` //判断整数k是否属于当前集合

❖ 如何具体实现？各接口的效率如何？

Bitmap : 思路

❖ 使用物理地址连续的一段空间，各元素依次对应于一个比特位

❖ class Bitmap {



private:

char * M; //以char (8比特) 为单位的比特位图

int N; //位图长度 (以sizeof(char)为单位)

public:

Bitmap(int n = 8) { M = new char[N = (n+7)/8]; memset(M, 0, N); }

~Bitmap() { delete [] M; M = NULL; } //析构

void set(int k); void clear(int k); bool test(int k); //ADT

};

Bitmap : 实现

```
❖ void set(int k)    { expand(k);          M[k >> 3] |= ( 0x80 >> (k & 0x07) ); }  
  
❖ void clear(int k) { expand(k);          M[k >> 3] &= ~( 0x80 >> (k & 0x07) ); }  
  
❖ bool test(int k)  { expand(k); return M[k >> 3] & ( 0x80 >> (k & 0x07) ); }  
  
❖ void expand( int k ) { //Bitmap[k]出界时扩容，分摊 $O(1)$   
    if ( k < 8 * N ) return; //仍在界内，无需扩容  
    int oldN = N; char * oldM = M;  
  
    M = new char[ N = ( ( 2 * k + 7 ) / 8 ) ]; //与向量类似，加倍策略  
  
    memcpy_s( M, N, oldM, oldN ); delete [] oldM;  
}
```