9.词典

（xb3）位图：快速初始化

邓俊辉

deng@tsinghua.edu.cn

O(1)初始化

- Bitmap的构造函数中，通过memset(M, 0, N)统一清零

  这一步只需O(1)时间？不，实际上仍等效于诸位清零，O(N) = O(n)！

- 尽管这并不会影响上例的渐进复杂度，但并非所有问题都是如此

- 有时，对于大规模的散列表，初始化的效率直接影响到实际性能

  例如： 第11章中bc[]表的构造算法，需要O(|Σ| + m) = O(s + m)时间

  若能省去bc[]表各项的初始化，则可严格地保证是O(m)

- 有时，甚至会影响到算法的整体渐进复杂度

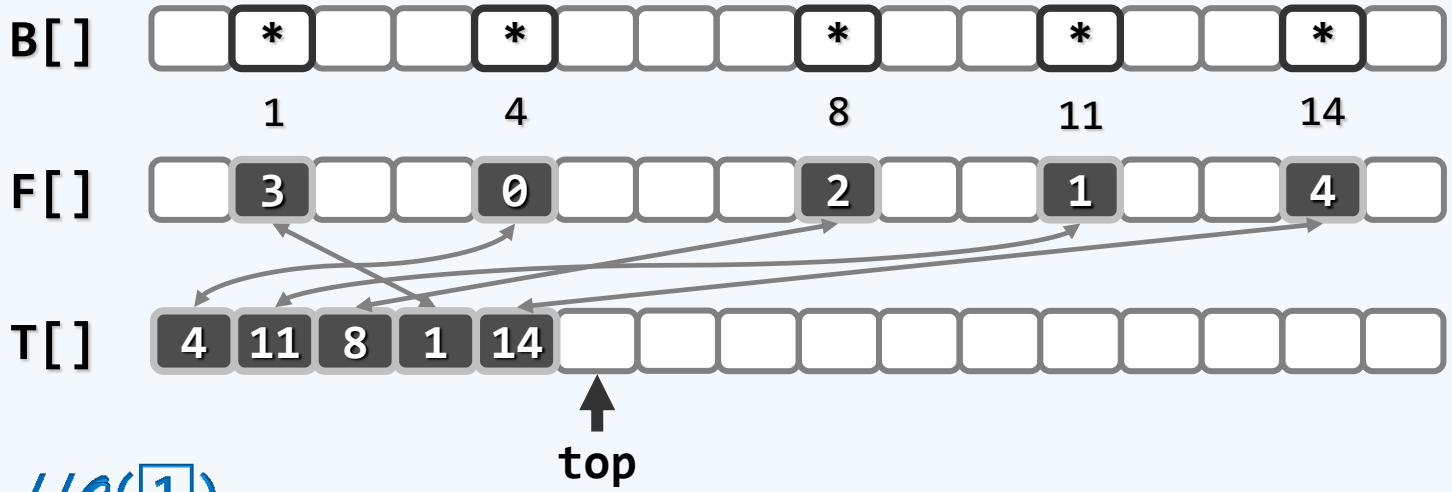
  例如，为从n = 10^8个32位整数中找出重复者，可仿造剔除算法... //但这里无需回收

  因此，若能省去Bitmap的初始化，则只需O(n)时间

O(1)初始化

❖ //[J. Hopcroft, 1974] 为B[m]增设一对等长的Rank型向量

```
Rank F[ m ]; //向量from
Rank T[ m ]; //向量（栈）to
Rank top = 0; //to的栈顶
//总体空间复杂度仍为O(m)
```



❖ `bool Bitmap::test( int k ) //O(1)`

```
{ return 0 <= F[ k ] && F[ k ] < top && T[ F[ k ] ] == k; }
```

❖ `void Bitmap::set( int k ) //O(1)`

```
{ if ( ! B.test( k ) ) { T[ top ] = k; F[ k ] = top++; } }
```

```
void Bitmap::clear( int k ) { /* 更为复杂，习题[2-34] */ } //O(1)
```