

第4章 矩阵数组与字符串

作业选讲

数据之法
结构之美
算法之道



zhuyungang@jlu.edu.cn



习题1

- 二维数组A中，每个元素的长度为3个字节，行下标从0到9，列下标从0到11，则连续存放该数组至少需要多少个字节？
- 10行12列，120个元素，每个元素3字节，所以共360字节

习题2

二维数组A有4行8列，下标从0开始，存储A的起始地址为2000，每个元素用相邻的4个字节存储，试计算：(1) 存储整个数组一共需要多少个字节。(2) 数组A的最后一个元素的起始地址。(3) 按行存储时，A[2][4]的起始地址。(4) 按列存储时，A[3][2]的起始地址。

(1) $4 \times 32 = 128$

(2) 2124

(3) 按行优先: $LOC(A[i][j]) = LOC(A[0][0]) + (i \times 8 + j) \times 4$

$LOC(A[2][4]) = 2000 + (2 \times 8 + 4) \times 4 = 2080$

(4) 按列优先: $LOC(A[i][j]) = LOC(A[0][0]) + (j \times 4 + i) \times 4$

$LOC(A[3][2]) = 2000 + (2 \times 4 + 3) \times 4 = 2044$

A[0][0]	A[0][1]	A[0][2]	A[0][3]	A[0][4]	A[0][5]	A[0][6]	A[0][7]
A[1][0]							
A[2][0]				A[2][4]			
A[3][0]		A[3][2]					



习题3

设模式字符串P=“abcabcacabca”，则该模式串的 $next$ 函数为

a	b	c	a	b	c	a	c	a	b	c	a
-1	0	0	0	1	2	3	4	0	1	2	3

习题4

已知主串s=“abcaabbabcbacbacba”，模式串p=“abcabaa”，写出模式串的失败函数，并由此画出KMP算法匹配的全过程。

第一趟匹配：

a	b	c	a	<u>a</u>	b	<u>b</u>	a	b	c	a	b	a	<u>a</u>	c	b	a	c	b	a
↑	↑	↑	↑	↑															
a	<u>b</u>	c	a	b															

第二趟匹配：

a	b	c	a	<u>a</u>	b	<u>b</u>	a	b	c	a	b	a	<u>a</u>	c	b	a	c	b	a
			↑	↑															
			a	<u>b</u>															

第三趟匹配：

a b c a a b b a b c a b a a c b a c b a

↑ ↑ ↑

a b c

第四趟匹配：

a b c a a b b a b c a b a a c b a c b a

↑

a

第五趟匹配：

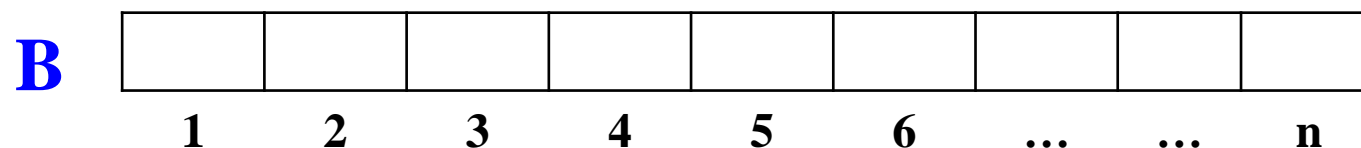
a b c a a b b a b c a b a a c b a c b a

↑ ↑ ↑ ↑ ↑ ↑

a b c a b a a

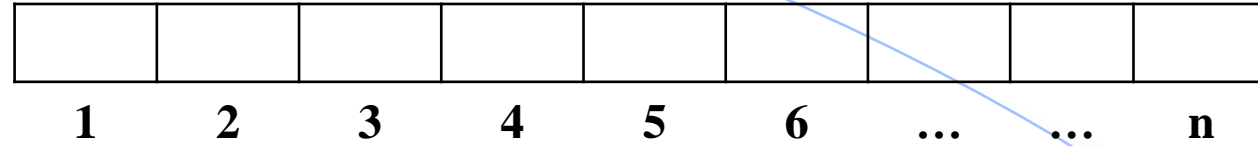
习题5

给定一个含 $n(n \geq 1)$ 个整数的数组A，请设计一个在时间上尽可能高效的算法，找出数组中未出现的最小正整数。例如，数组 $\{-5, 3, 2, 3\}$ 中未出现的最小正整数是1；数组 $\{1, 2, 3\}$ 中未出现的最小正整数是4。并给出算法的时间复杂度和空间复杂度。【2018年考研题全国卷，13分】



数组B标示 $1 \dots n$ 之间的元素在A中是否出现：

- $B[i]=1$ 表示 i 在数组A中出现，
- $B[i]=0$ 表示 i 没在数组A中出现。



遍历数组A，当前元素 $k=A[i]$:

- 如果 $1 \leq k \leq n$, $B[k]=1$.
- 否则，不操作.

时间复杂度 $O(n)$
空间复杂度 $O(n)$

```
int FindMissMin(int A[ ],int n){  
    int *B = new int[n+1];  
    for(int i=1; i<=n; i++) B[i]=0;  
    for(int i=0; i<n; i++)  
        if(A[i]>=1 && A[i]<=n) B[A[i]]=1;  
    for(int i=1; i<=n; i++)  
        if(B[i]==0) break;  
    delete B; return i;  
}
```