**云南大学信息学院计算机科学与技术专业**

**《数据结构》第4-5章 串和数组 作业 任课教师：孔兵**

**姓名：白文强 学号：20191060064 得分：**

1．描述以下概念的区别：空格串与空串。

答：空格是一个字符，其ASCII码值是32。空格串是由空格组成的串，其长度等于空格的个数。空串是不含任何字符的串，即空串的长度是零。

2. 已知n阶下三角矩阵A（即当i<j时，有aij=0），按照压缩存储的思想，可以将其主对角线以下所有元素(包括主对角线上元素 )依次存放于一维数组B中，请写出从第一列开始采用列序为主序分配方式时在B中确定元素aij的存放位置的公式。

答：n阶下三角矩阵元素A[i][j]（1<=i,j<=n,i>=j）。第1列有n个元素，第j列有n-j+1个元素，第1列到第j-1列是等腰梯形，元素数为(n+(n-j+2)(j-1)/2，而aij在第j列上的位置是为i-j+1。所以n阶下三角矩阵A按列存储，其元素aij在一维数组B中的存储位置k与i和j的关系为：

k=(n+(n-(j-1)+1)(j-1)/2+(i-j+1)=(2n-j)(j-1)/2+i

3.假设一个准对角矩阵：



按以行为主序存于一维数组B[0..4m-1]中，写出由一对下标（i, j）求k的转换公式。

i为奇数时：k = i+j-2；

i为偶数时：k = i+j-1；

k = 2(i/2)+i-1;

4. 设有三对角矩阵(aij)n\*n， 将其三条对角线上的元素逐行地存于数组B(1:3n-2)中，使得B[k]=aij，求：

（1）用i,j表示k的下标变换公式；

（2）用k表示i,j的下标变化公式。

(1)k = (i-2)\*3 + 2 + (j-i+1)+1 = 2i + j -2;

(2)i = k/3 + 1;

j = k–2\*(k/3 + 1) + 2;

(1)k = (i-2)\*3 + 2 + (j-i+1)+1 = 2i + j -2;

(2)i = k/3 + 1;

j = k–2\*(k/3 + 1) + 2;

**5.**试设计一个C算法（或C程 序）：用单链表作存储结构，以回车为结束标志，输入一个任意长度的字符串；然后判断该字符串是否为“回文”（正向读和反向读时，串值相同的字符串称为“回 文”），输出信息“Yes”或“No”；最 后删除字符串并释放全部空间。例如：

若输入“ABCD12321DCBA” 是回文，则输出“Yes”;

若输入“ABCD123DCBA”， 不是回文，则输出“No”。

要求：定义相关数据类型，不得使用数组（顺序表）作字符串的存诸结构和辅助存储空间。假定字符串的长度为n ,试分析上述算法的时间复杂度。

**void** **Compare(LinkList** **former,LinkList** **latter)**

**{**

**while(former** **!=** **NULL)**

**{**

**if(former->data** **!=** **latter->data)**

**//出现一处不相等，就必不是回文串**

**{**

**printf("No\n");**

**break;**

**}**

**former** **=** **former->next;**

**latter** **=** **latter->next;**

**}**

**if(former** **==** **NULL)**

**{**

**printf("YES\n");**

**}**

**}**

**LinkList** **Div\_ReverseList(LinkList** **\*L)//快慢指针，将链表逆转一半分成两个等长链表**

**{**

**LinkList** **lslow** **=** **\*L;//慢指针**

**LinkList** **lfast** **=** **(\*L)->next;//快指针**

**LinkList** **temp** **=** **NULL;**

**if(lslow->next** **==** **NULL)**

**{**

**return** **lslow;**

**}**

**while(lfast->next** **!=** **NULL** **&&** **lfast->next->next** **!=** **NULL)**

**{**

**\*L** **=** **lslow->next;**

**lslow->next** **=** **temp;**

**temp** **=** **lslow;**

**lslow** **=** **\*L;//慢指针移动一格**

**lfast** **=** **lfast->next->next;//快指针移动两格**

**}**

**if(lfast->next** **==** **NULL)**

**{**

**\*L** **=** **lslow->next;**

**lslow->next** **=** **temp;**

**return** **lslow;**

**}//此时，链表分成了以lslow和\*L为头结点的两个单链表**

**else**

**{**

**LinkList** **t** **=** **(\*L)->next;**

**\*L** **=** **lslow->next->next;**

**lslow->next** **=** **temp;**

**free(t);//把中间的节点释放掉**

**return** **lslow;**

**}//此时链表分成了以temp和\*L为头结点的两个链表**

**}**

**typedef** **struct** **LNode**

**{**

**char** **data;**

**struct** **LNode** **\*next;**

**}** **LinkNode,** **\*LinkList;**

**//结构体定义**

**Div\_ReverseList函数中，核心语句\*L** **=** **lslow->next; 执行n/2次**

**Compare函数中，核心语句former** **=** **former->next; 平均执行n/4次**

**所以时间复杂度为O(n)；**