循环链表编程综合练习：  
1.（a）修改教材图4.8的代码，实现循环单链表  
（b）修改图4.14 的代码，实现循环双链表  
2.Josehus问题的描述：数字1，2，…，n，n个人顺时针围着一个圆圈坐着，每个人都持有一个密码（正整数）。选择一个正整数开头作为报数m的限制，从第一人开始按顺时针方向从一个报数开始，报数m停止报数。谁报了m出界，他的密码作为新的m值，从下一个人开始按顺时针方向重新报出一个数字，一直到所有人都列到现在为止。设计一个程序使用循环单链表，按出列顺序打印每个数字。要求提交完整的可运行的源程序和运行结果截图。

第五次课 课前小测  
1.顺序表中插入第i个元素需要移动多少个元素？

i>=1的话，移动n-i+1  
2.顺序表中删除1个元素需要平均移动多少个元素？

(n-1)2/4  
3.写一个算法，要求查找单链表中是否有值为K的元素。

第4次课 课前小测  
1.以下算法的时间复杂度为\_\_\_\_\_\_\_\_\_\_\_。  
void fun( int n)  
{ int i=1;  
 while(i<=n)  
      i=i\*2;  
}  
A. O(n)  B. O(n^2)  C. O(nlogn)  D. O(logn)  
  
2. 以下算法的时间复杂度为\_\_\_\_\_\_\_\_\_\_\_。  
void fun( int n)  
{ int i,j,x=0;  
 for(i=1;i<n;i++)  
for(j=n;j>=i+1;j--)  
   x++;  
}  
A. O(n)  B. O(n^2)  C. O(nlogn)  D. O(n^3)  
  
3.以下算法的时间复杂度为\_\_\_\_\_\_\_\_\_\_\_。  
void fact( int n)  
{ if(n<=1) return 1;  
else return n\*fact(n-1);  
}  
A. O(n)  B. O(n^2)  C. O(nlogn)  D. O(logn)  
  
4.只使用线性表ADT定义里提供的函数，设计一个算法，按从前往后的顺序打印出线性表中所有的元素。

第6次课 课前小测  
1.写出构成Array-base list 类型的私有成员数据项，画一个非空的顺序表和空顺序表的示意图  
2.写出构成single linked list 类型的私有成员数据项，画一个非空的单链表和空链表的示意图

循环链表编程综合练习：

1.（a）修改教材图4.8的代码，实现循环单链表

（b）修改图4.14 的代码，实现循环双链表

2.Josehus问题的描述：数字1，2，…，n，n个人顺时针围着一个圆圈坐着，每个人都持有一个密码（正整数）。选择一个正整数开头作为报数m的限制，从第一人开始按顺时针方向从一个报数开始，报数m停止报数。谁报了m出界，他的密码作为新的m值，从下一个人开始按顺时针方向重新报出一个数字，一直到所有人都列到现在为止。设计一个程序使用循环单链表，按出列顺序打印每个数字。要求提交完整的可运行的源程序和运行结果截图。

栈stack的编程练习：

1. 利用栈实现单链表的逆置操作（写成外部函数，在单链表类型定义中如果需要，可以增加一个修改当前节点值的操作函数setvalue）

2. 编程实现将一个十进制数转换成d进制数

栈的应用编程练习2：

题目描述：

给定一个字符串，里边可能包含“()”、"{}"、“[]”三种括号，请编写程序检查该字符串的括号是否成对出现。

输出：

true：代表括号成对出现并且嵌套正确，或字符串无括号字符。

false：未正确使用括号字符。

第9次课 课前小测

1. Assume a series of values as 1,2,3,4,5,6 passes a stack, an impossible output sequence is ( ) .

A. 2,4,3,5,1,6 B. 3,2,5,6,4,1

C. 1,5,4,6,2,3 D. 4,5,3,6,2,1

2. When a recursive algorithm is transformed into a no recursive algorithm, a structure( ) is generally used.

3.The RPN of the infix expression A\*(B+C)/(D-E-F) is ( )

4.Let Q be a non-empty queue, and let S be an empty stack. Using only the stack and queue ADT functions and a single element variable X, write an algorithm to reverse the order of the elements in Q.

1. 假设当1、2、3、4、5、6传递一个堆栈时，一个不可能的输出序列是()。

A. 2、3、4、5、5、5、5、6、6

C. 1、5、4、6、2、3

2. 将递归算法转换为非递归算法时，一般使用structure()。

3.中缀表达式A\*(B+C)/(D-E-F)的RPN为()

4.设Q是一个非空队列，设S是一个空堆栈。只使用stack和queue ADT函数和一个元素变量X，编写一个算法来颠倒Q中元素的顺序。

队列的编程练习：

1. 修改图4.26顺序队列的定义，使用一个独立的布尔成员记录队列是否为空，而不必在数组中留一个空位置。

2. 双端队列（deque）是一种队列，但是它的元素同时可以从队首和队尾进行插入和删除操作，编写一个双端队列，分别用数组和链表实现。（可以参考用一个数组实现2个栈的实现）.

3.回文是指一个字符串从前面读和从后面读都一样。仅使用若干栈和队列ADT函数及若干个int类型和char类型的变量，编写一个算法，判断一个字符串是否为回文。假设字符串从标准输入设备一次读入一个字符，算法的输出结果应为true或者false。

编程练习：

4.编程实现使用循环队列作为存储结构解决约瑟夫问题。

Josehus问题的描述：数字1，2，…，n，n个人顺时针围着一个圆圈坐着，每个人都持有一个密码（正整数）。选择一个正整数开头作为报数m的限制，从第一人开始按顺时针方向从一个报数开始，报数m停止报数。谁报了m则出列，他的密码作为新的m值，从下一个人开始按顺时针方向重新报出一个数字，一直到所有人都出列为止。设计一个程序使用循环队列，按出列顺序打印每个数字。要求提交完整的可运行的源程序和运行结果截图。

第11次课 课前小测.

1.为了便于检索，字典里面存储的数据一般是什么形式：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.假设有三维数组A 6\*8\*9，每个元素用相邻的8个字节存储，存储器按字节编址。已知A的起始存储位置（基地址）为1000，计算：

（1）数组A的体积（即存储量）；

（2）数组A的最后一个元素a578的开始地址；

（3）按行存储时，元素a436的开始地址；

（4）按列存储时，元素a517的开始地址。

第12次课 课前小测

1.假设对对称矩阵M 8\*8使用压缩存储，每个元素用相邻的4个字节存储，存储器按字节编址。已知M的起始存储位置（基地址）为2000，计算：

（1）矩阵M需要的存储空间；

（2）矩阵M的最后一个元素a77的开始地址；

（3）按行存储时，元素a54的开始地址；

（4）按行存储时，元素a36的开始地址。

2.假设将下三角矩阵M 10\*10（第一个元素为a11）压缩存储在一维数组A中，计算：

（1）矩阵M需要存储的元素个数；

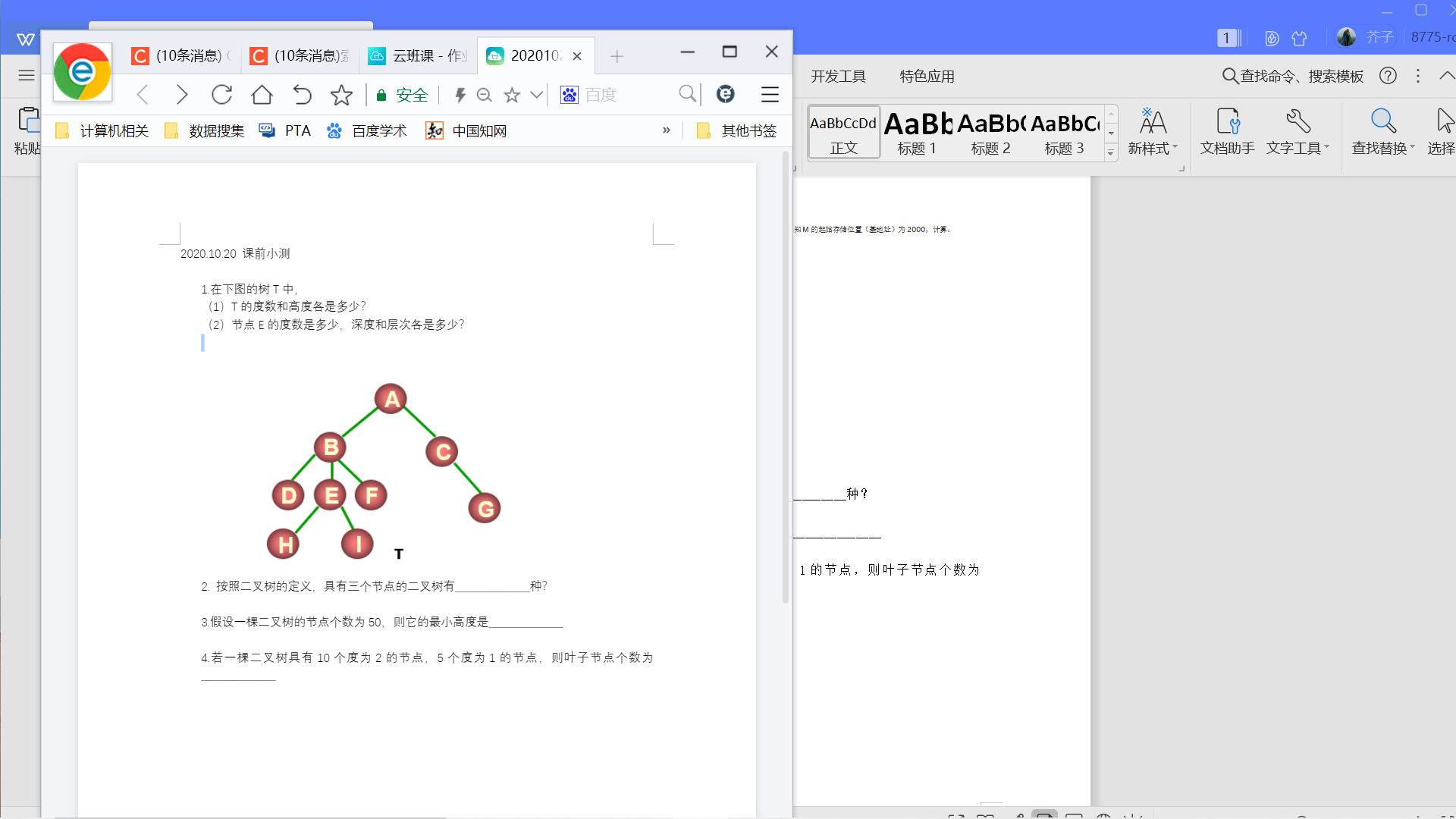
（2）按行存储时，元素a63在A中存储位置；

（3）按行存储时，元素a39在A中存储位置。

1.在下图的树T中，

（1）T的度数和高度各是多少？

（2）节点E的度数是多少，深度和层次各是多少？



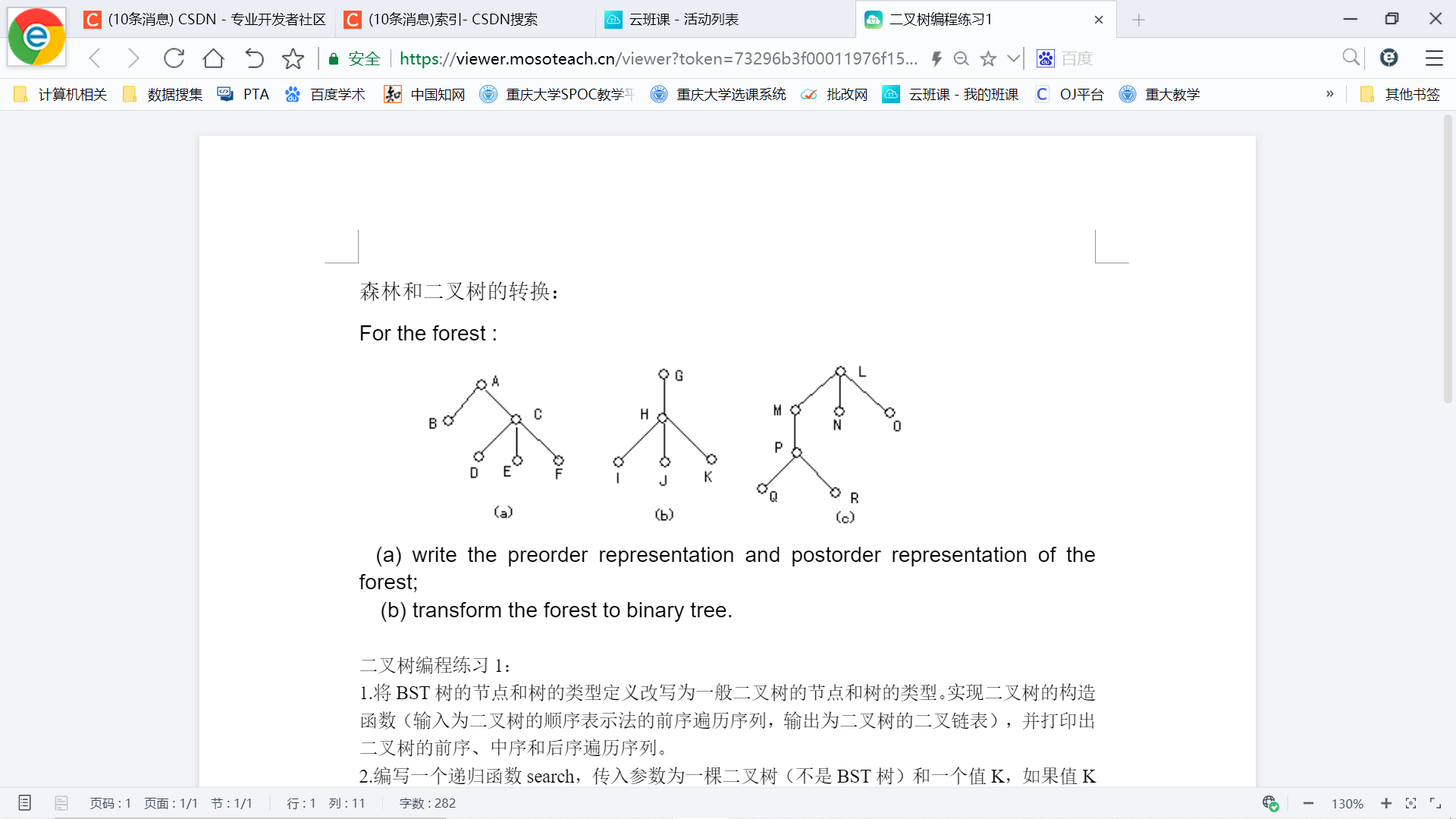
2. 按照二叉树的定义，具有三个节点的二叉树有\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_种？

3.假设一棵二叉树的节点个数为50，则它的最小高度是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.若一棵二叉树具有10个度为2的节点，5个度为1的节点，则叶子节点个数为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

森林和二叉树的转换：

For the forest :

(a) write the preorder representation and postorder representation of the forest;

(b) transform the forest to binary tree.

(a)编写森林的预定表示和后预定表示;

(b)将森林转化为二叉树。

二叉树编程练习1：

1.将BST树的节点和树的类型定义改写为一般二叉树的节点和树的类型。实现二叉树的构造函数（输入为二叉树的顺序表示法的前序遍历序列，输出为二叉树的二叉链表），并打印出二叉树的前序、中序和后序遍历序列。

2.编写一个递归函数search，传入参数为一棵二叉树（不是BST树）和一个值K，如果值K在二叉树中则返回true，否则返回false

3.编写一个算法，传入参数为二叉树根节点的指针，输出为层次遍历序列。

以上编程题都要求程序能够编译运行（节点和树的类型定义最好写成头文件，3道题都可以用），同时上传源程序文件和运行结果截图。

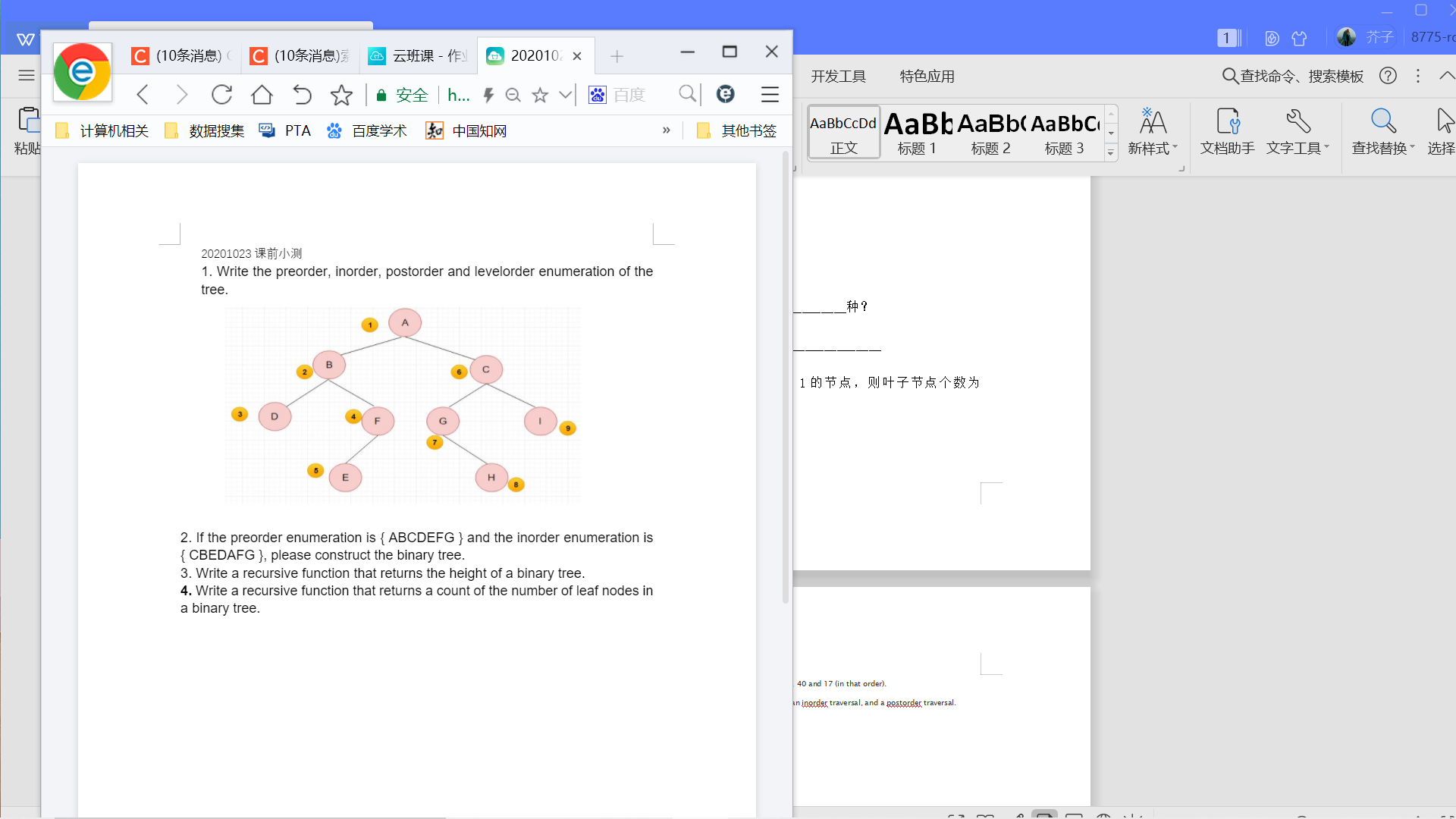
BST树编程练习：

实现BST树类型定义，假设一个BST在节点中保存整数值，完成下列任务：

1.编写一个递归函数，返回其所有节点的数值之和。

2.编写一个递归函数smallcount，给定一个BST的根和值K，返回值小于或等于K的节点数目。函数smallcount应该尽可能少的访问BST的节点。

3.编写一个递归函数printRange，给定一个BST的根，一个较小的值和一个较大的值，按照顺序打印出介于这2个值之间的所有节点。函数printRange应该尽可能少的访问BST的节点。



第16次课 课前小测

1. (a)Draw the BST that results from inserting the values 30, 20, 35, 18, 26, 5, 28, 24, 32, 40 and 17 (in that order).

(b) Show the enumerations for the tree of (a) that result from doing a preorder traversal, an inorder traversal, and a postorder traversal.

(c) Draw the BST that results from deleting the value 32 from the BST of (a).

(d) Draw the BST that results from deleting the value 18 from the BST of (a).

(e) Draw the BST that results from deleting the value 5 from the BST of (a).

(f) Draw the BST that results from deleting the value 30 from the BST of (a).

1. (a)绘制插入值30、20、35、18、26、5、28、24、32、40和17所得到的BST(按此顺序)。

(b)显示(a)树的枚举，它是通过执行前序遍历、顺序遍历和后序遍历而产生的。

(c)绘制从(a)的BST中删除32的结果BST。

(d)绘制从(a)的BST中删除18的结果BST。

(e)绘制从(a)的BST中删除值5所产生的BST。

(f)绘制从(a)的BST中删除值30所产生的BST。

第18次课 课前小测：

1. （1）根据下列元素关键字值的输入顺序用快速建堆法建立大根堆：45，28，63，12，98，76， 37，50

（2）在（1）所建立的大根堆上插入88

（3）在（2）所建立的大根堆上删除第一个节点

（4）在（3）所建立的大根堆上删除节点63

2. （1）根据下列元素关键字值的输入顺序用快速建堆法建立小根堆：45，28，63，12，98，76， 37，50

（2）在（1）所建立的小根堆上插入08

（3）在（2）所建立的小根堆上删除第一个节点

（4）在（3）所建立的小根堆上删除节点28

Complete the implementation of the Huffman coding tree, building on the code presented in Section 5.6. Include a function to compute and store in a table the codes for each letter, and functions to encode and decode messages. This project extended to support ﬁle compression. To do so requires adding two steps: (1) Read through the input ﬁle to generate actual frequencies for all letters in the ﬁle; and (2) store a representation for the Huffman tree at the beginning of the encoded output ﬁle to be used by the decoding function. If you have trouble with devising such a representation, see Section 6.5.

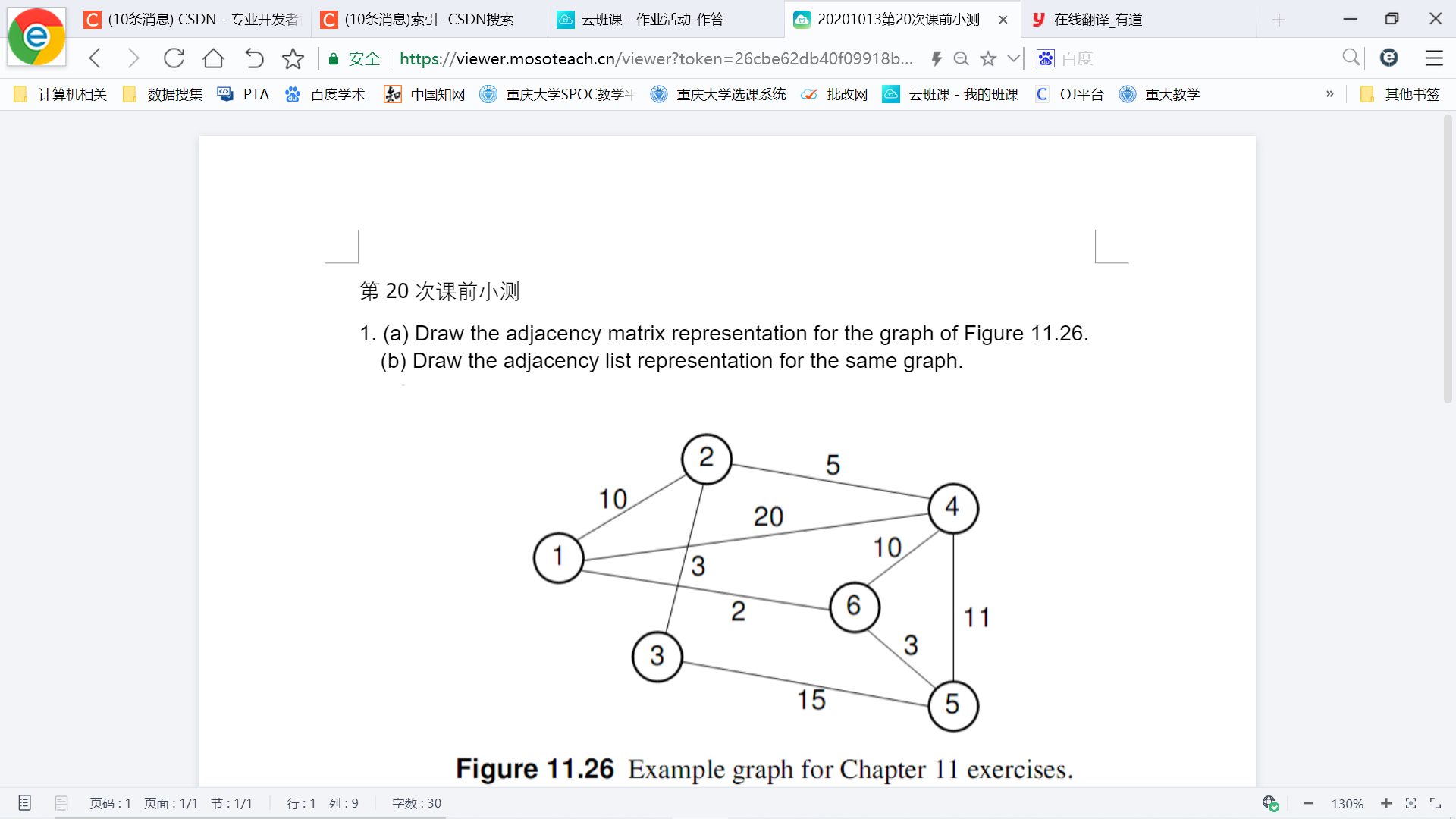
在第5.6节介绍的代码基础上，完成Huffman编码树的实现。包括计算和存储每个字母的代码的函数，以及编码和解码消息的函数。这个项目扩展到支持文件压缩。要做到这一点，需要添加两个步骤:(1)读取输入文件，以生成文件中所有字母的实际频率;以及(2)在编码输出文件的开始处存储霍夫曼树的表示，以供解码函数使用。如果设计这样的表示有困难，请参阅第6.5节。

(a) Draw the adjacency matrix representation for the graph of Figure 11.26.

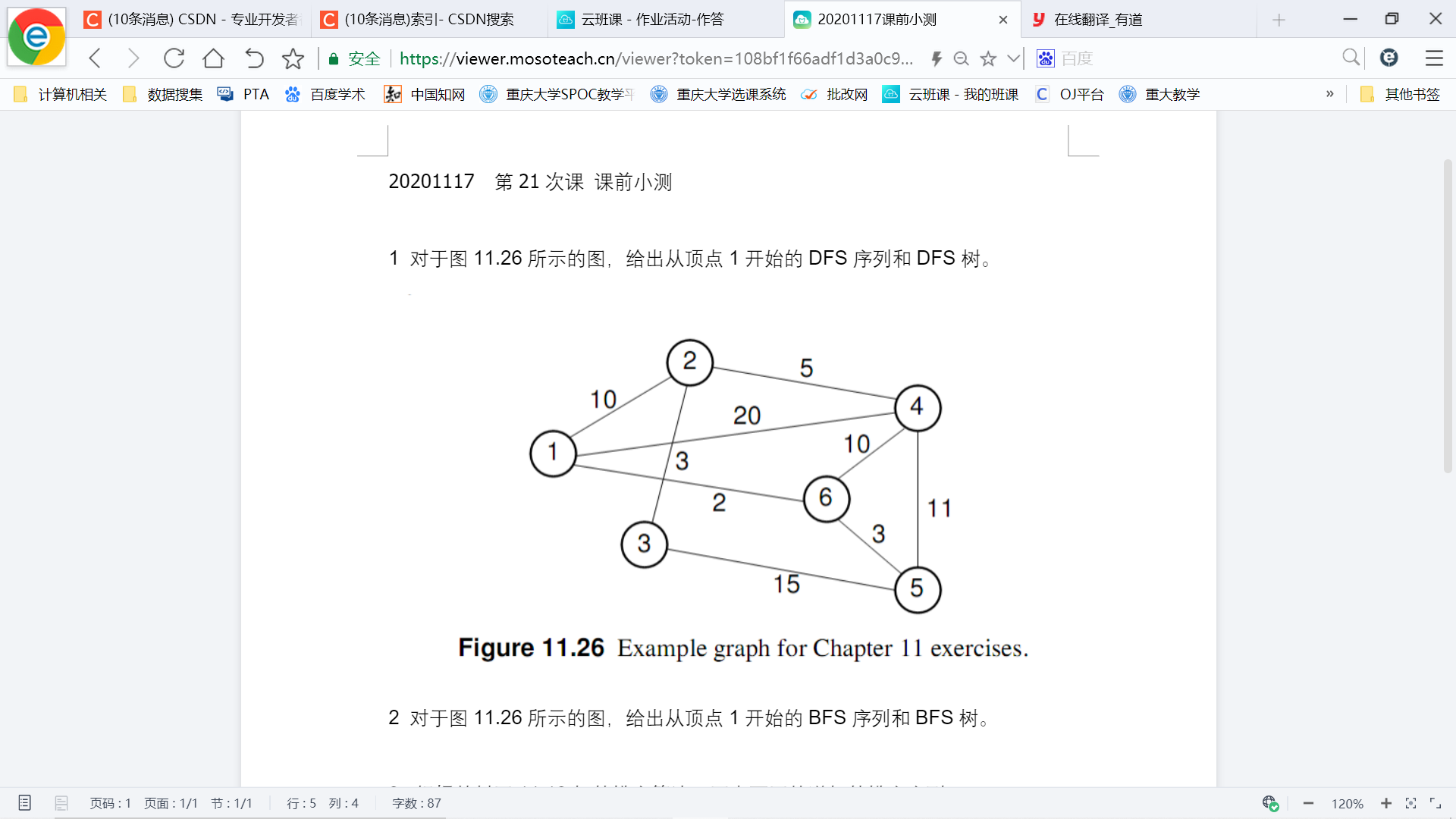
(b) Draw the adjacency list representation for the same graph.

(a)画出图11.26图的邻接矩阵表示。

(b)绘制同一图的邻接表表示法。

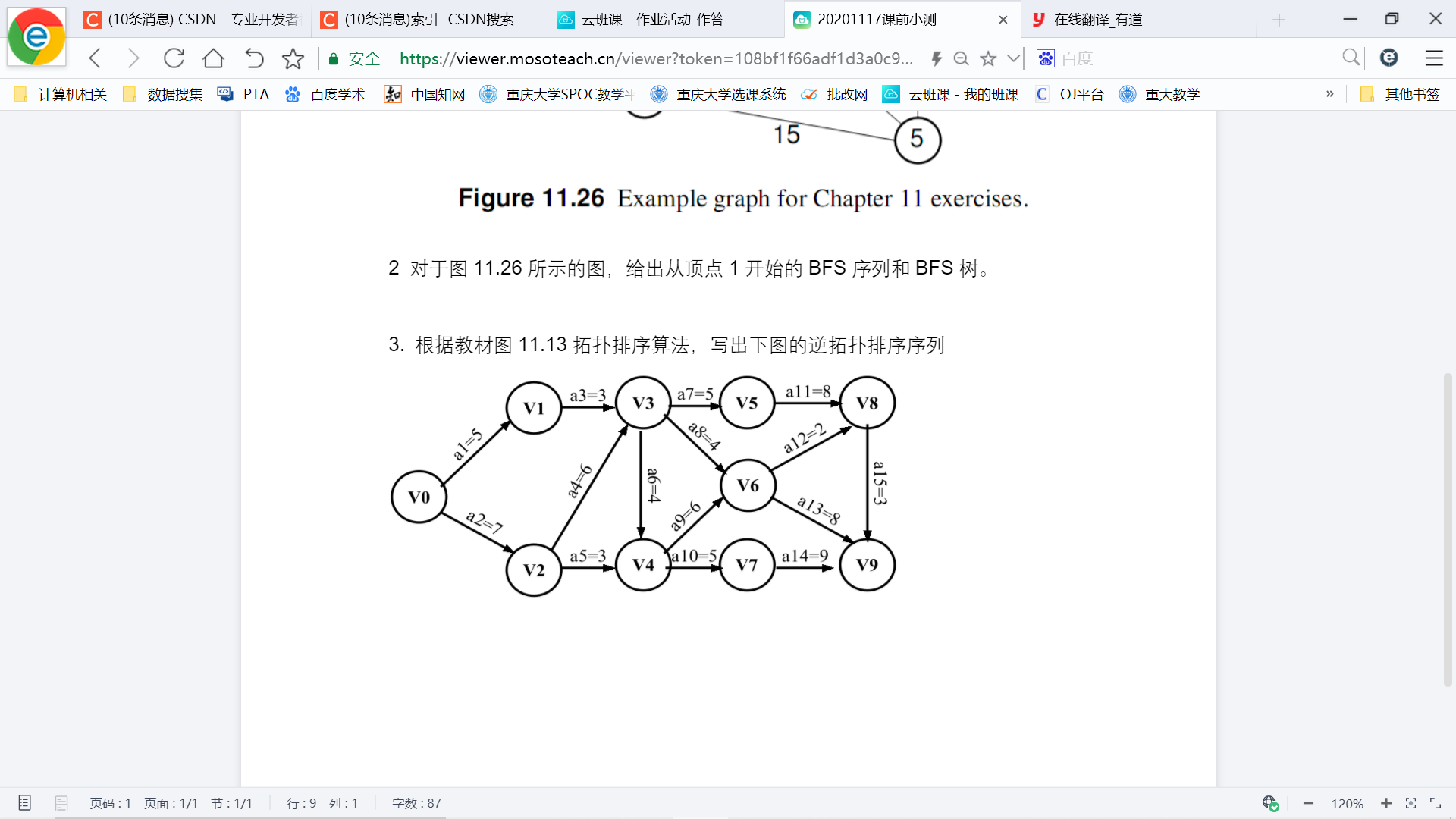


1 对于图11.26所示的图，给出从顶点1开始的DFS序列和DFS树。



2 对于图11.26所示的图，给出从顶点1开始的BFS序列和BFS树。

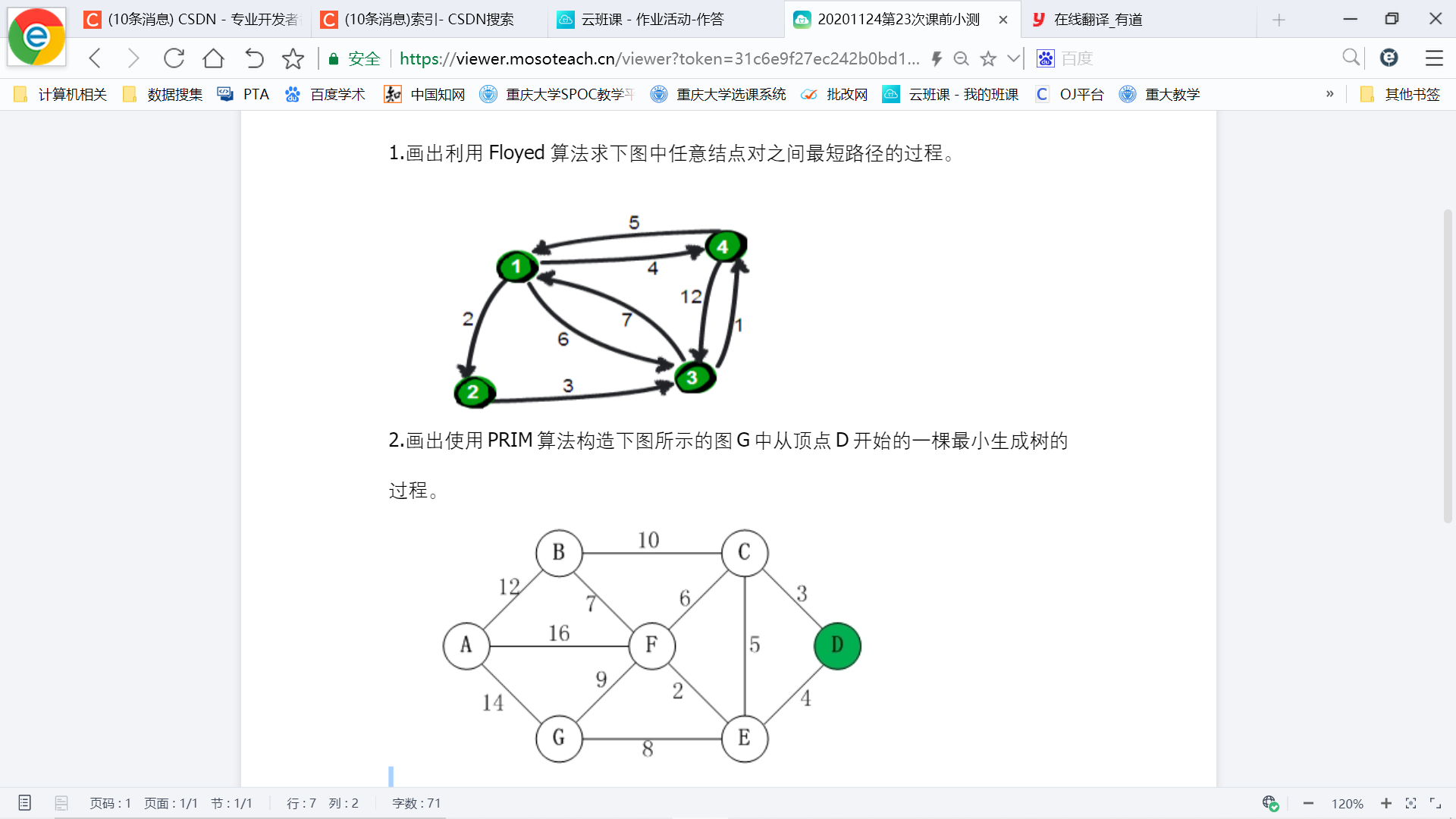
3. 根据教材图11.13拓扑排序算法，写出下图的逆拓扑排序序列



第25次课 课前小测

1.给定一组关键字序列{16,69,71,28,6,35,8,45,12,31,55, 28’}，写出用希尔排序算法（增量的选择按照教材的方法）进行非递减排序时，每一趟排序的结果。

2.给定一组关键字序列{12,65,48,23,34,5,2,41,37,19,52, 23’}，分别写出用归并排序和快速排序进行非递减排序时排序的过程。



第27次课 课前小测

给定一组关键字序列{12,65,48,23,34,5,2,41,37,19,52, 28}，写出用教材上数组基数排序算法进行非递减排序时排序的过程。