数据结构与算法平时作业讲解提纲

1. 线性结构反映结点间逻辑关系是_____的，非线性结构反映结点间逻辑关系是_____的。

   ANS: 一对一 ；一对一或一对多或多对多

   分析: 线性结构的数据之间关系是一对一的关系，除了第一个最后一个元素和最后一个元素之外，每个元素都有且仅有一个前驱和后续节点。常见的线性结构有线性表，堆栈，队列，数组，串。

   非线性的数据之间关系可能是一对一，一对多或多对多的关系，每个元素可能有零个，一个，多个以上的前驱和后续节点。常见的非线性数据结构有树（二叉树）和图。

2. 在树型结构中，树根结点没有_____结点，其余每个结点有且只有_____个前趋驱结点；叶子结点没有_____结点；其余每个结点的后续结点可以_____。

   ANS: 前驱节点 ；一 ；后续节点 ；有叶子节点

   分析： 概念。

3. 下面程序段的时间复杂度是_____。

   ```
   for(i=0;i<n;i++)
   for(j=0;j<n;j++)
   A[i][j]=0
   ```

   ANS: $O(n^2)$

分析： 嵌套循环用乘法法则，外层时间复杂度 n，内层时间复杂度 n，总的事件复杂度$O(n*n=n^2)$

4. 下面程序段的时间复杂度是_____。

```
i=s=0;
while(s<n)
{
    i++;
    s+=i;
}
```

ANS：$O(\sqrt{n})$

分析：

第一句 O(1)。

假设循环进行了 x 次。当满足$S=1+2+3+\cdots+x\geq 0$ 退出循环，该循环进行了 x 次，粗略估计当$\frac{x(x+1)}{2}\geq n$退出循环，即$x\geq\sqrt{n}$退出循环，取最高项$O(\sqrt{n})$。

5. 下面程序段的时间复杂度是_____。

```
s=0;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
s+=B[i][j];
sum=s;
```

ANS： $O(n^2)$

分析：第一句 O(1)，中间的循环 $O(n^2)$，最后一句 O(1)，取最高项 $O(n^2)$。

6. 下面程序段的时间复杂度是_____。

```
i=1;
while(i<=n)
i=i*3;
```

ANS：$O(\log_3 n)$

分析：

第一句 O(1)。

假设循环进行了 x 次，当满足$\mathbf{x} = \mathbf{x_{n-1}} * \mathbf{3} > \boldsymbol{n}$退出循环，即$\mathbf{x} \geq \mathbf{log_3}\, \boldsymbol{n}$退出循环，时间复杂度$\sqrt{\mathbf{n}}$。

7. 算法时间复杂度的分析通常有两种方法，即_____和_____的方法，通常我们对算法求时间复杂度时，采用_____方法。Why?

ANS: 事前估计；事后统计；事前估计；事后统计容易受软件硬件纯等环境因素印象，有时候容易掩盖算法的优势。

分析：概念。

8. 在顺序/链表中插入和删除一个结点平均需要移动多少个结点?具体的移动次数取决于什么因素?

ANS:顺序表插入一个节点需要平均移动$\frac{\mathbf{n}}{\mathbf{2}}$个节点，删除一个节点需要平均移动$\frac{\mathbf{n-1}}{\mathbf{2}}$个节点，具体移动次数取决于该循序表长度和需要进行插入或删除位置，越靠近顺序表尾节点越

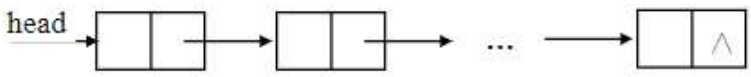
移动次数越少。链表不需要移动节点，但是需要得到上个节点指向该节点的指针（单向链表），因此需要额外的记录。

分析：顺序表最坏的情况是第一个元素就是要添加或者删除的元素，这样会将原来所有的元素向前或向后移动一个位置。

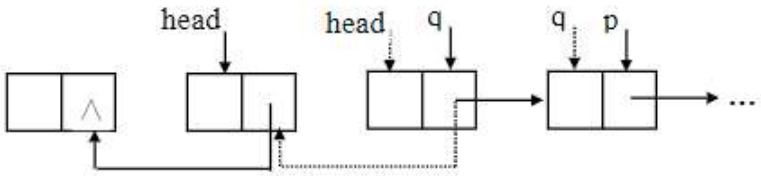
9. 写一算法实现单链表的逆置。（不带头结点）

ANS&解析：

定义每个单链表数据形式如下，其中 data 为节点数据，next 下个节点的地址。头节点无数据，只有记录首节点（第一个有效数据节点）的地址。



(a) 单链表初始状态



(b) 第三个结点逆置

单链表逆置示意图

1) 方法 1

```
typedef structNode
{
    elemtype data;
    structNode *next;
} LinkList;
void contray(LinkList *head)
{//将head单链表中所有结点按相反次序链接
    LinkList*p,*q;
```

```
    p=head;  //p指向未被逆序的第一个结点,初始时指向原表头结点
    head=NULL;
    while(p!=NULL)
    {
        q=p; //q指向将被逆序链接的结点
        p=p->next;
        q->next=head;
        head=q;
    }
}
```

2) 方法 2

```
typedef structNode
{
    elemtype data;
    structNode *next;
} LinkList;
void reverse(LinkList *head)
{
    LinkList *now=head; //当前节点为头节点
    LinkList *prev=NULL; //头节点上个节点为空
    LinkList *latter; //先不定义下个节点
    while(now!=NULL)
    {
        latter=now->next; //取得当前节点下个节点的值
        now->next=prev; //当前节点指向上个节点;
        prev=now;  //上个节点指向当前的节点
        now=latter; //现在节点变成下个要逆序的节点
    }
    head = prev; //更新头节点位置
}
```

10. 线性表、栈和队列都是_____结构，可以在线性表的_____位置插入和删除元素；对于栈只能在_____位置插入和删除元素；对于队列只能在_____位置插入元素和在_____位置删除元素

ANS: 线性；任意位置；栈顶；队尾；队首

分析：概念。

11. 设有一空栈，现有输入序列 1，2，3，4，5，经过

push,push,pop,push,pop,push,push 后，输出序列是

_____。

ANS&分析：

1 进栈　栈内元素 栈顶—1—栈底

2 进栈　栈内元素 栈顶—2—1—栈底

2 出栈　栈内元素 栈顶—1—栈底

3 进栈　栈内元素 栈顶—3—1—栈底

3 出栈　栈内元素 栈顶—1—栈底

4 进栈　栈内元素 栈顶—4—1—栈底

5 进栈　栈内元素 栈顶—5—4—1—栈底

最后出栈 5—4—1

12. 无论对于顺序存储还是链式存储的栈和队列来说，进行插入或删除运算的时间复杂度均相同为_____。

ANS: O(n)

分析：

元素处在线性表表尾：线性表查询元素最坏 O(n)，移动数据最好 O(1)，

元素处在线性表表头：查询最好 O(1)，移动数据最 O(n)；

因此线性表的时间复杂度 O(n)。

链表不需要移动数据，查询最坏 O(n)。

13. 什么是队列的上溢/假溢出现象？一般有几种解决方法，

简述之。

ANS&分析：上溢是队尾指针 rear 已经到达存储空间大小 maxnum。此时无法在新元素加入队列，发生上溢现象。假溢出则是队列中有足够的空间但是无法将元素插入队列，一般是队列的存储结构或操作方式选择不当，可以用循环队列解决。

解决队列尚上溢出可以建立一个足够大内存空间，但是这样会造成空间使用低，浪费空间，所以我们要使用其他方案。

避免队列的假溢出可以使用以下几种方法

1） 采用移动元素的方法。若队列中空间足够，每当一个新元素入队，就将队列中已有的元素向对头一个位置。

2） 每当删除一个队头元素，则将其他元素向前移动，使队头指针 front 指向队列的第一个位置。

3） 使用循环队列。将队头、队尾看作一个首位相接的循环队列，使用循环数组实现，同样满足队列的特性，做插入和删除运算遵循“先进先出”的原则，队尾进，队头出。

14. 两个字符串相等的充要条件是_____和_____。

ANS：字符串长度相等；对应位置元素相同。

分析：概念。

15. 空串是指_____，空格串是指_____。

ANS: 字符串中没有任何元素；字符串仅有一个元素，该元素是“ ”。

分析：概念。

16. 设定串采用顺序结构，写出对串 s1 和串 s2 比较大小的算法。串值大小按字典排序（升序）方式，返回值等于-1，0 和 1 分别表示 s1<s2，s1=s2 和 s1>s2。

ANS&分析：

题目的意思如下：

两者长度相同，元素相同　　返回 0
两者长度不同，在共有长度（短串）S1[i]>S2[i]　　返回 1
两者长度不同，在共有长度（短串）S1[i]<S2[i]　　返回-1
两者长度不同，共有长度元素（短串）相同，下个元素 S1[i]>S2[i]　　返回 1
两者长度不同，共有长度元素（短串）相同，下个元素 S1[i]<S2[i]　　返回-1

算法描述如下：

```
#define MAXLEN 256

struct strnode
{
    char data[MAXLEN];
    int len;
}SeqString;//定义顺序串类型

int strcmp(SeqString s1,SeqString s2)//比较串s1和串s2的大小
{
    //先求出最短的长度
    int minlen;
    if(s1.len>s2.len)
        minlen=s2.len;
    else
        minlen=s1.len;
    int i=0;
    for(;i<minlen;i++)
    {
```

```
        if(s1.data[i]<s2.data[i])
            return (-1);
        else if(s1.data[i]>s2.data[i])
            return (1);
    }
    if(s1.data[i]==s2.data[i])
        return(0);//s1=s2
    else if(s1.data[i]<s2.data[i])
        return(-1);//s1<s2
    else
        return(1);//s1>s2
  }
```

17. 设有一个长度为 s 的字符串，其字符顺序存放在一个一维数组的第 1 至第 s 个单元中（每个单元存放一个字符）。现要求从此串的第 m 个字符以后删除长度为 t 的子串，m<s，t<(s-m)，并将删除后的结果复制在该数组的第 s 单元以后的单元中，请设计此算法。

ANS&分析：

```
int delete(char r[],int s,int t,int m)
{
    m++;//变成要删除字串第一个位置
    s++;//变成要复制到的第一个位置
    for(int i=0;i<t;i++)
    {
        r[s]=r[m+t];
        s++;
        m++;
    }
    return 1;
}
```

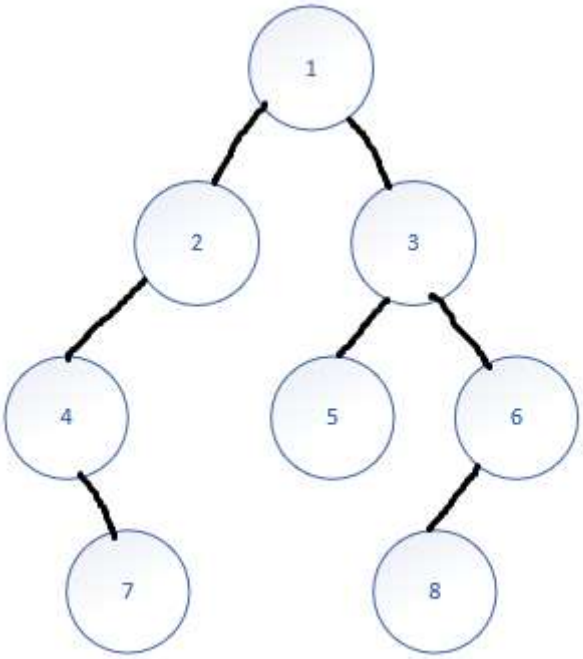
18. 已知一棵二叉树的先序遍历序列和中序遍历序列分别为 1,2,4,7,3,5,6,8 和 4,7,2,1,5,3,8,6，请画出这棵二叉树，然后写出该树的后序遍历序列。

ANS&分析：

前 1 2 4 7 3 5 6 8

中 4 7 2 1 5 3 8 6

后 7 4 2 5 8 6 3 1



19. 已知一棵二叉树的后序遍历序列{7,4,2,5,8,6,3,1}和中序遍历序列{4,7,2,1,5,3,8,6}，请画出这棵二叉树，然后写出该树的先序遍历序列。

ANS&分析：见上题。

20. 直接插入排序方法的稳定性和时间复杂度是？已知待排关键字序列为{8，6，1，0，2，5，9，7}，请写出直接插入排序每一趟的排序结果（降序）。

ANS&分析：直接插入排序是稳定排序方法，时间复杂度 $O(n^2)$

1) 8

2) 6 8

3) 1 6 8

4) 0 1 6 8

5) 0 1 2 6 8

6) 0 1 2 5 6 8

7) 0 1 2 5 6 8 9

8) 0 1 2 5 6 7 8 9

9) 0 1 2 5 6 7 8 9

21. 待排关键字序列为{8，6，1，0，2，5，9，7}，请写出冒泡排序（升序）每一趟的排序结果。

ANS&分析：

冒泡排序原理：

1．比较相邻的元素。如果第一个比第二个大，就交换他们两个。
2．对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。这步做完后，最后的元素会是最大的数。
3．针对所有的元素重复以上的步骤，除了最后一个。
4．持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。

冒泡排序伪代码：

```
function bubble_sort (array, length) {
    var i, j;
    for(i from 0 to length-1){
        for(j from 0 to length-1-i){
            if (array[j] > array[j+1])
                swap(array[j], array[j+1])
        }
    }
}
```

输入 8 6 1 0 2 5 9 7

1) 6 8 1 0 2 5 9 7

6 1 8 0 2 5 9 7

6 1 0 8 2 5 9 7

6 1 0 2 8 5 9 7

6 1 0 2 5 8 9 7

跳过

6 1 0 2 5 8 7 9

（共循环 7 次）

2) 1 6 0 2 5 8 7 9

1 0 6 2 5 8 7 9

1 0 2 6 5 8 7 9

1 0 2 5 6 8 7 9

跳过

1 0 2 5 6 7 8 9

3) 0 1 2 5 6 7 8 9

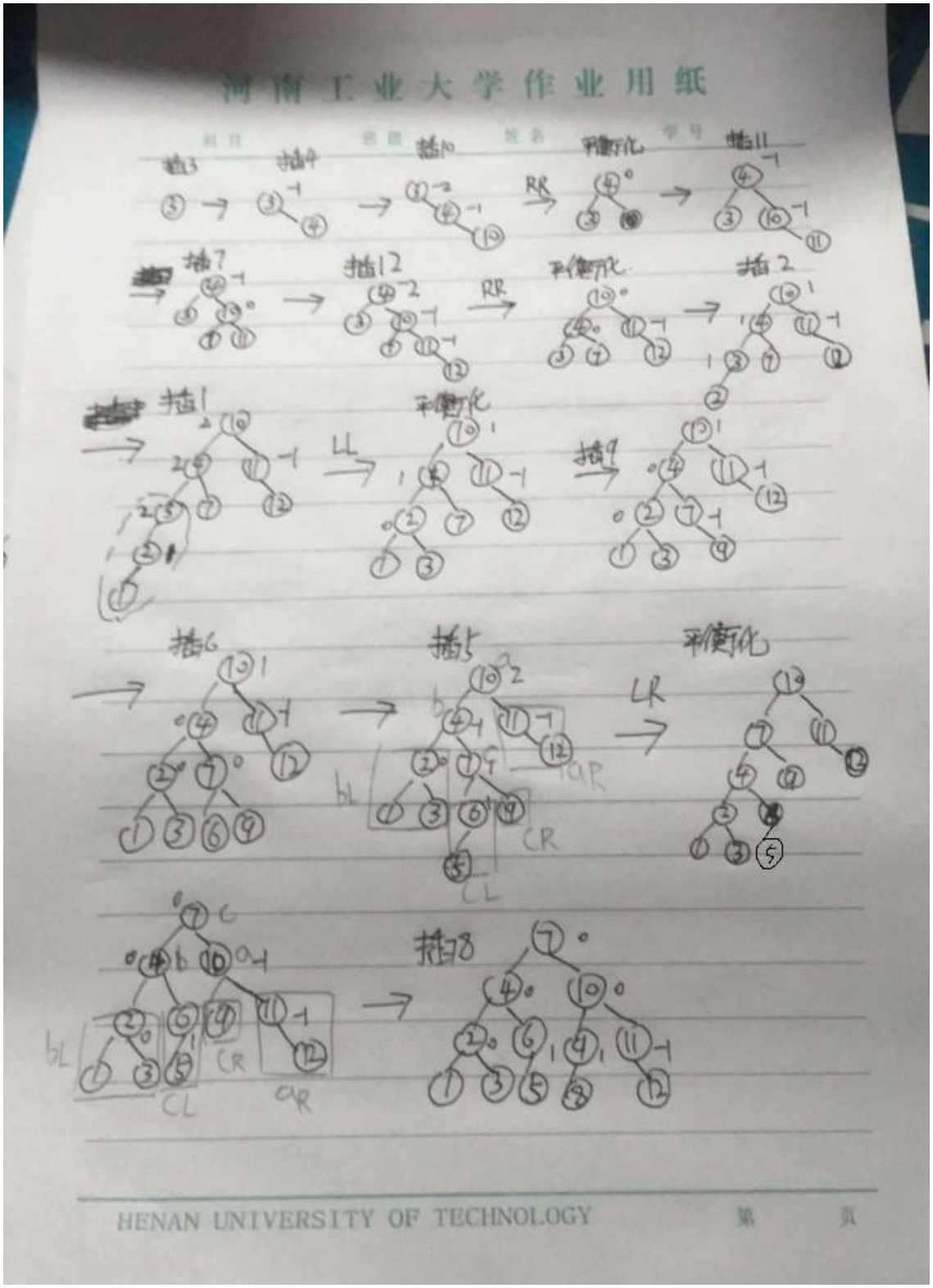
跳过

4) 0 1 2 5 6 7 8 9

退出循环

22. 已知某数值序列为(3,4,10,11,7,12,2,1,9,6,5,8)，请画出对应的二叉排序树，其是否为 AVL 树，如不是，请用一系列图形详细描述出生成其对应的 AVL 树的过程（并标明失衡原因如 RR)，并计算平均查找长度 ASL。

ANS&分析：

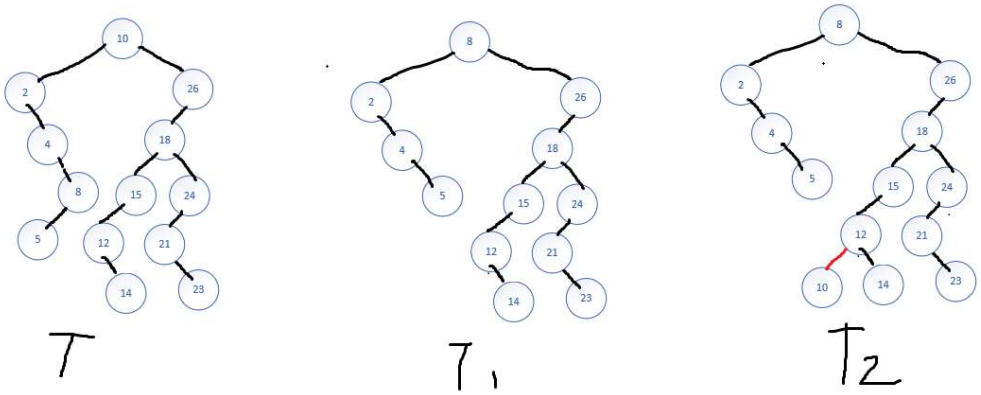


平均查找长度：

$$\textit{平均查找长度} = \frac{1}{\textit{节点总数}}(\Sigma(\text{层数} * \text{该层节点数}))$$

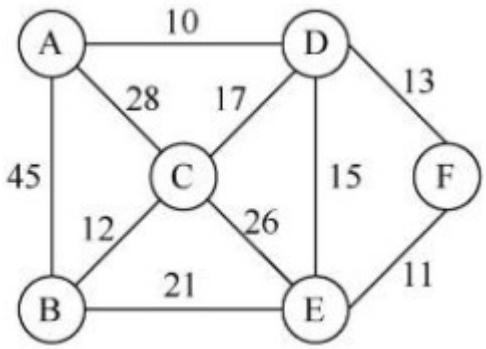
$$\frac{37}{12}=\frac{1}{12}(1*1+2*2+3*4+4*5)$$

23. 将关键字(10,2,26,4,18,24,21,15,8,23,5,12,14)依次插入到初态为空的二叉排序树中，请画出所得到的树 T；然后画出删除 10 之后的二叉排序树 T1；若再将 10 插入 T1 中得到的二叉排序树 T2 是否与 T1 相同?请画出 T2，并写出先序遍历序列。

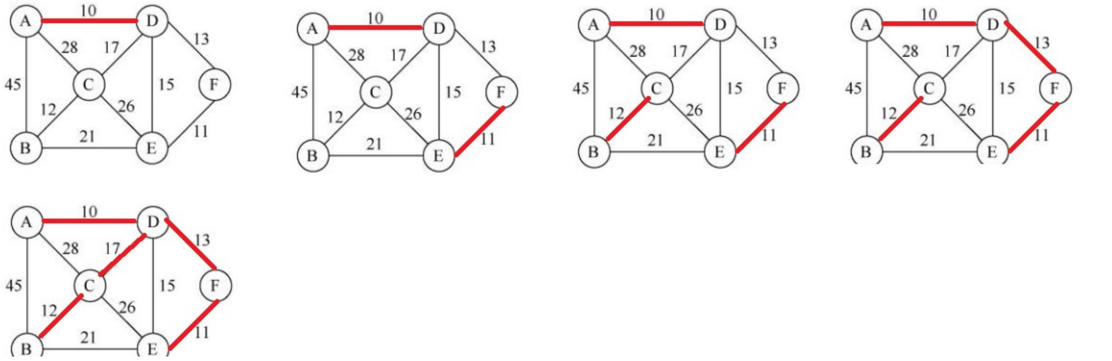


先序遍历 8，2，4，5，26，18，15，12，10，14，24，21，23

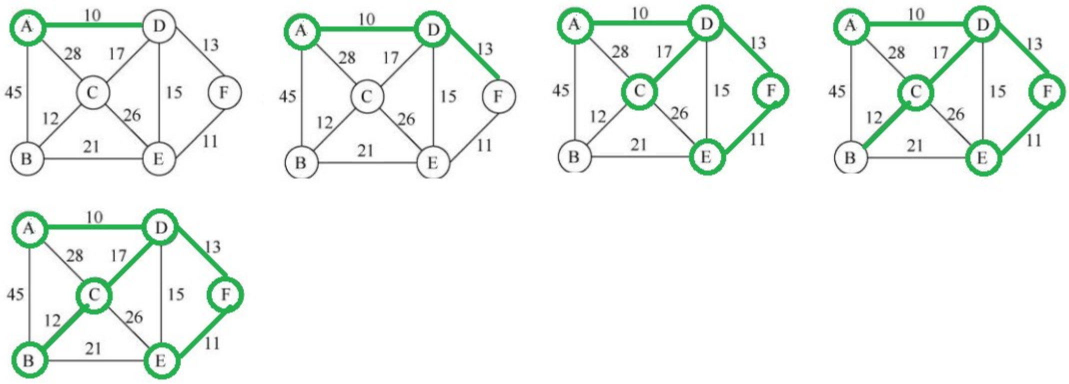
24. 对于图所示的带权无向图。请按照 Prime 算法画出从顶点 C 开始构造最小生成树的过程。请按照克鲁斯卡尔算法画出构造最小生成树的过程。



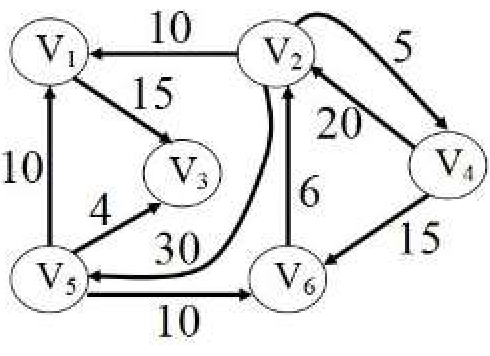
克鲁斯卡尔



Prime



25.　请对图示带权有向图 G，用 Dijkstra 算法求从顶点 6 到其余各顶点的最短路径，请先写出邻接矩阵的初始值，并在下表中补足描述数组 Dist 和 pre 的各分量的变化过程。



顶点 \ 步骤		1	2	3	4	5	S
初态	Dist	∞	6	∞	∞	∞	{6}
	pre	6	6	6	6	6	
1	Dist	16	6	∞	11	36	{6,2}
	pre	2	6	6	2	2	
2	Dist	16	6	∞	11	36	{6,2,4}
	pre	2	6	0	2	2	
3	Dist	16	6	31	11	36	{6,2,4,1}
	pre	2	6	1	2	2	
4	Dist	16	6	31	11	36	{6,2,4,1,3}
	pre	2	6	1	2	2	
5	Dist	16	6	31	11	36	{6,2,4,1,3,5}
	pre	2	6	1	2	2	