

圣才考研网 www.100exam.com

参 考 答 案

一、

1. $n-1$

2. 5

3. $\frac{n}{2} n-1$

4. 16

网 www.100exam.com

5. 25 26

9. $2e$

6. $2n-1$

10. $n+1$

7. 顺序 是有序的

8. $O(n\log n)$ $O(d(n+rd))$

二、

1. C

2. B

3. B

4. B

5. A, C

6. D

7. C

三、

1. \times

2. \checkmark

3. \times

4. \checkmark

5. \checkmark

6. \times

7. \checkmark

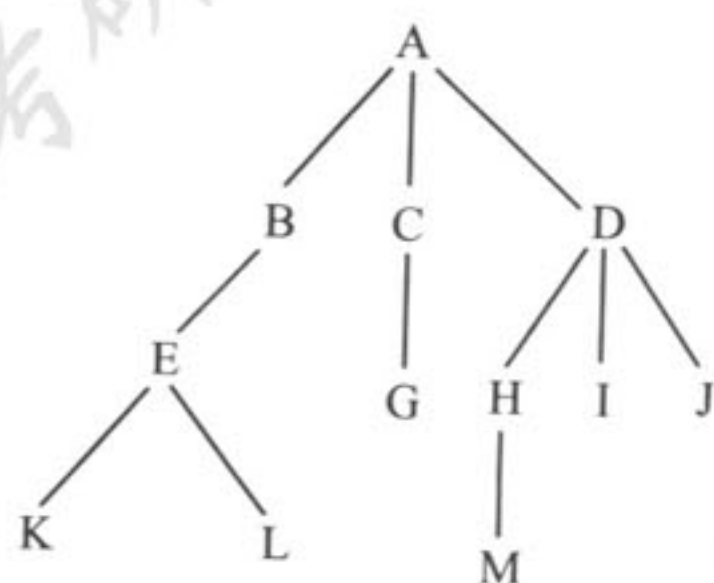
8. \checkmark

9. \checkmark

10. \checkmark

四、

1.

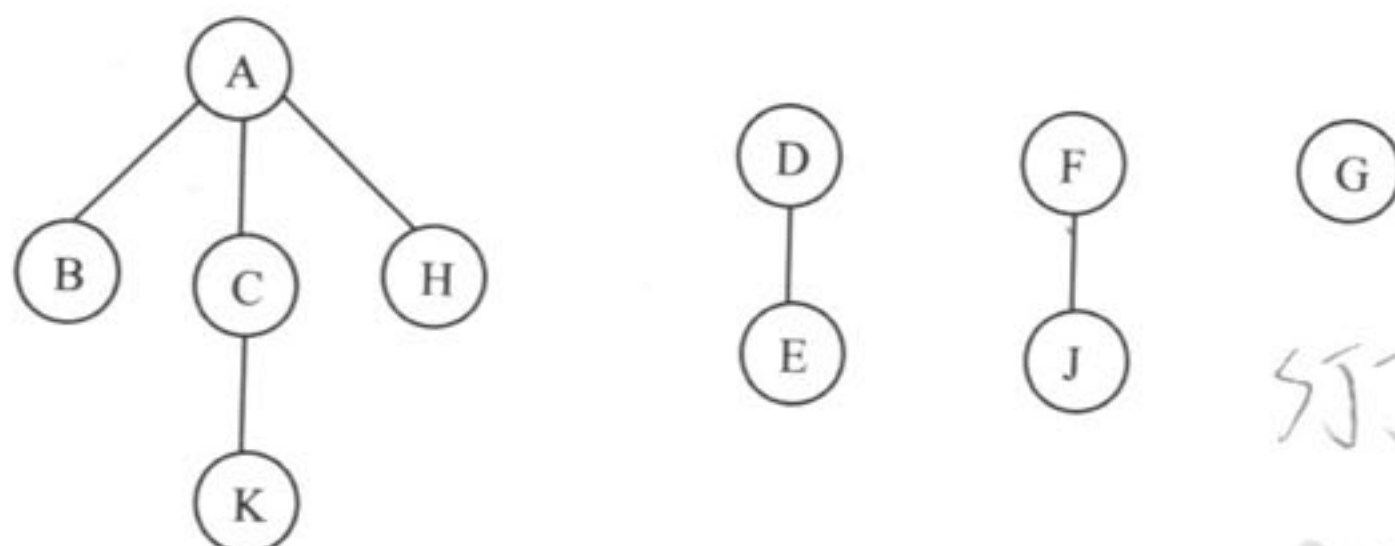


2. 对于具有同一排序码的多个记录来说,若采用的排序方法使排序后记录的相对次序不变,则称此排序方法是稳定的,否则称为不稳定的。

例: 12, 11, 18, 15, 11, 经过快速排序后得到的序列为 11, 11, 12, 15, 18
其中11与11的次序与原记录的次序不相同,所以快速排序是不稳定的排序。

3. 由集合上一偏序得到该集合的一个全序的操作称之为拓扑排序。

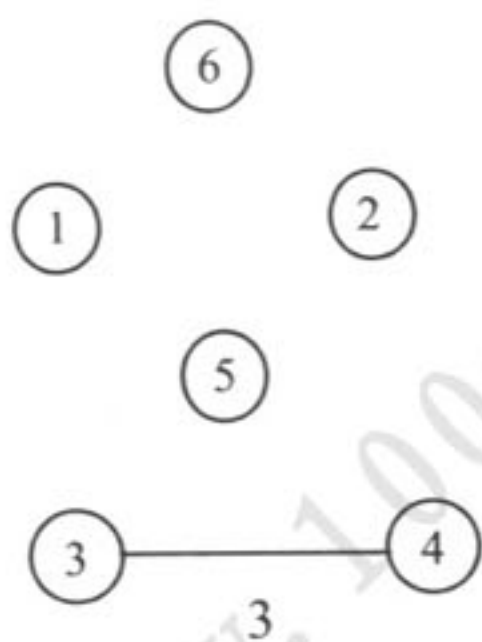
五、



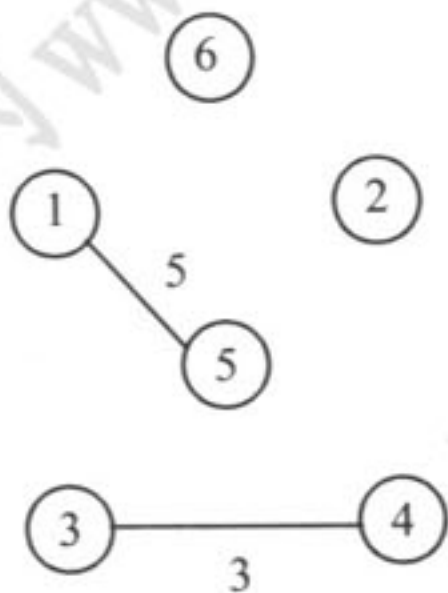
六、

该无向最小生成树可分为五步:

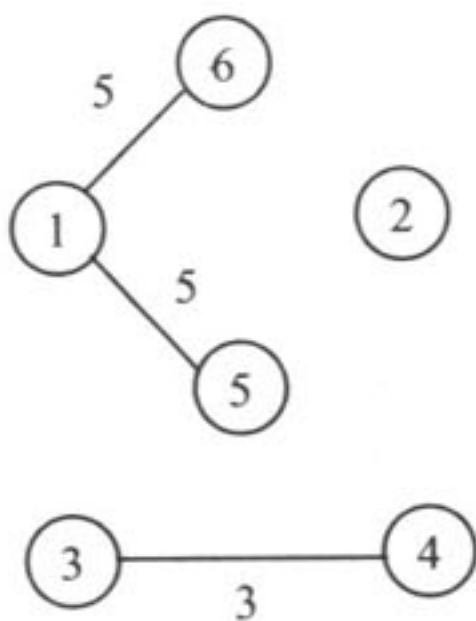
(1)



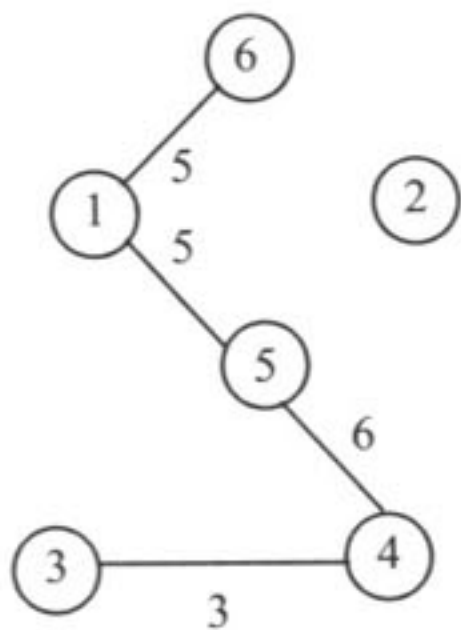
(2)



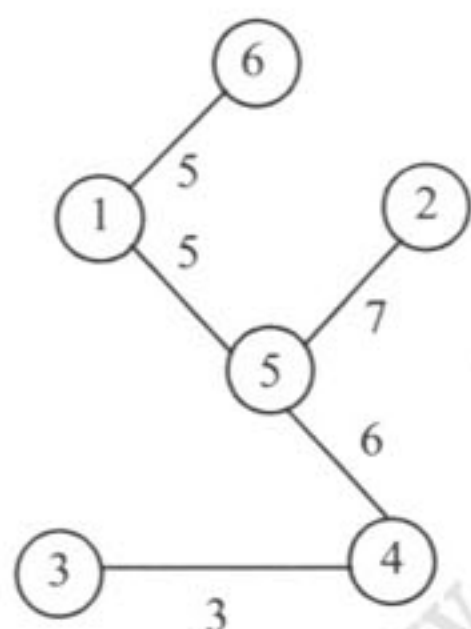
(3)



(4)



(5)



七、

设图的邻接矩阵为 $G[n][n]$ （针对无向图的）

定义邻接表结点的类型为：struct edgenode{
int adjvex;
edgenode * next;
}

typedef edgenode * adjlist[n]; //定义adjlist为存储n个表头指针的数组类型

将邻接矩阵表示转换为邻接表表示的算法如下：

```
void matrixtolist (int G[ ][ ], adjlist GL, int n)
{
    edgenode *p, *q;
    for (int i=0;i<n;i++) GL[i]=Null;
    for (int i=0;i<n;i++)
        for (int j=0;j<n;j++){
            if (G[i][j]!=0) p=(edgenode *) malloc (sizeof(edgenode));
            p->adjvex=j;
            p->next=Null;
            if (GL[i]== Null){GL[i]=p;q=p;}
            else {q->next=p;
                    q=p;
            }
        }
}
```

八、

定义树的结点类型为：typedef struct node {
ElemType data;
struct node *lchild;
struct node *rchild;
}BTree;

定义静态全局变量a[3]，其中a[0]、a[1]、a[2]分别用来存储度数分别为0、1、2结点的个数：

```
static int a[3]
```

算法如下:

```
void countnode (BTree *b)
```

```
{
    if (b==Null) return;
    if (b->lchild==Null && b->rchild==Null) a[0]=a[0]+1;
    if (b->lchild!=Null && b->rchild!=Null) a[2]=a[2]+1;
    else a[1]=a[1]+1;
    countnode (b->lchild);
    countnode (b->rchild);
}
```

九、(9分)

以邻接表的存储结构为例, 邻接表的数据类型如下:

```
typedef struct {
    int adjvex;           //该弧所指向的顶点的位置
    struct ArcNode * nextarc; //指向下一条弧的指针
    InfoType info;        //该弧的相关信息
} ArcNode;
typedef struct Vnode {
    Vertex data;          //顶点信息
    ArcNode *firstarc;    //指向第一条弧
} VNode;
typedef VNode AdjList [maxVertexNum]; // AdjList是邻接表类型
typedef struct {
    AdjList adjlist;      //邻接表
    int n, e;             //图中顶点数n和边数e
} ALGraph;               //图类型
```

算法如下:

```
static int visit [n];
void dfs (ALGraph *g, int v);
int dfscount (ALGraph *g)
{
    int i, j; j=0;
    for (i=0; i<g->n; i++)
        if (visit[i]==0){
            j++;
            dfs (g, i);
        }
}
void dfs (ALGraph *g, int v)
{
    ArcNode *p;
    visited[v]=1;
    p=g->adjlist[v], firstarc;
    while (p!=Null) {
        if (visited[p->adjvex]==0)
            dfs (g, p->adjvex);
        p=p->nextarc;
    }
```

```

    }
}

```

十、(8分)

```

void transfer (int a[ ], int n)
{
    int b[n];
    int k=0;
    int j=n-1;
    for (int i=0; i<n; i++) {
        if (a[i]%2==0) {
            a[k]=a[i];
            k++;
        } else {a[j]=a[i];
            j--;
        }
        for (int i=0; i<n; i++)a[i]=b[i]
    }
}

```

十一、(10分)

定义结点的结构为: struct Node {
 ElemType Data;
 struct node *next;
 }

定义栈的结构为: struct Stuck {
 Node * base;
 Node * top;
 }

定义队列的结构为: struct Queue {
 Node * front;
 Node * tail;
 }

设 m 个连续单元的数组为 $b[m]$, 定义全局数组static int $a[m]$ 用以标识 m 个单元中各个单元是否已被占用: $a[i]=1$ 表示已占用; $a[i]=0$ 表示未被占用。

```

void insertstack (struct stack&S, ElemType elemtype)
{
    for (int i=0;i<m;i++)
        if (a[i]==0) break;
    if (i==m) {
        count<<"There is no space"<<end;
        return;
    }
    a[i]=1;
    Node *p;
    P=&b[i];
    p->data=elemtype;p->next=NULL;
    if (S.BASE==Null) {
        S.base=p;
    }
}

```

```

        S.top=p;
    }
else
{
    p->next=top;
    S.top=p;
}
}
void insert Queue (struct Queue &Q,  ElemType elemtype)
{
    for (int i=0;i<m;i++)
        if (a[i]==0)break;
    if (i==m){
        count<<"There is no space"<<end;
        return;
    }
    a[i]=1;
    Node *p;
    p=&b[i];
    p->data=elemtype; p->next=NULL;
    if (Q.front==Null) {Q.front=p;
        Q.top=p;
    }else {
        Q.top->next=p;
        Q.top=p;
    }
}
}

```