# 4.6

void Prim(int n, int start)

{

    int i,j,x,t,ans=0;

*for* (i=1;i<=n;i++)

    {

        low[i]=v[start][i];

        point[i]=start;

    }

    point[start]=0;

*for* (i=1;i<n;i++)

    {

        t=INF;

        x=0;

*for* (j=1;i<=n;i++) *if* (point[j]&&low[j]<t)

        {

            x=j;

            t=low[j];

        }

        ans+=t;

*for* (j=1;j<=n;j++) *if* (point[j]&&v[x][j]<low[j])

        {

            low[j]=v[x][j];

            point[j]=x;

        }

    }

}

# 4.10

## （1）



## （2）



## （3）



## （4）



# 4.26

void floyd(int *n*)

{

    for (int k=1;k<=n;k++)

    {

        for (int i=1;i<=n;i++)

        {

            for (int j=1;j<=n;j++) a[i][j]=min(a[i][j],a[i][k]+a[k][j]);

        }

    }

}

int mid\_point(int *n*)

{

    floyd(n);

    int lowp=INF;

    for (int i=1;i<=n;i++)

    {

        b[i]=INF;

        for (int j=1;j<=n;j++) b[i]=min(b[i],a[j][i]);

        lowp=(b[i]<b[lowp]?i:lowp);

    }

    return lowp;

}

# 4.30

void DFS\_non\_recur(*Point* \**head*, *Point* \**start*)//邻接表-深搜非递归

{

    Setclear(head);

    Stack sta;

    sta.push(start);

    Node \*x,\*first,\*last=NULL;

    while (!sta.empty())

    {

        x=**new** Node();

        if (sta.get\_top()==start) first=x;

        x->add(sta.pop(),&last);

        x->data->vis=true;

        for (Point::Edge \*e=x->data->last\_edge;e!=NULL;e=e->next\_edge)

            if (e->terminal->vis==false) sta.push(e->terminal);

    }

    Print\_List(first);

}

# 4.44

## （1）

VE[1]=0     VL[1]=0

VE[2]=5     VL[2]=9

VE[3]=6     VL[3]=6

VE[4]=12    VL[4]=12

VE[5]=15    VL[5]=15

VE[6]=16    VL[6]=19

VE[7]=16    VL[7]=16

VE[8]=19    VL[8]=19

VE[9]=21    VL[9]=21

VE[10]=23   VL[10]=23

## （2）（3）

        E   L   L-E

(1,2)   0   4   4

(1,3)   0   0   0

(2,4)   5   9   4

(3,4)   6   6   0

(3,5)   6   12  6

(4,5)   12  12  0

(4,6)   12  15  3

(4,7)   12  12  0

(5,7)   15  15  0

(5,8)   15  15  0

(6,10)  16  19  3

(7,9)   16  16  0

(8,9)   19  19  0

(9,10)  21  21  0

## （4）

