Cac thuat toan to mau

Dan nhap

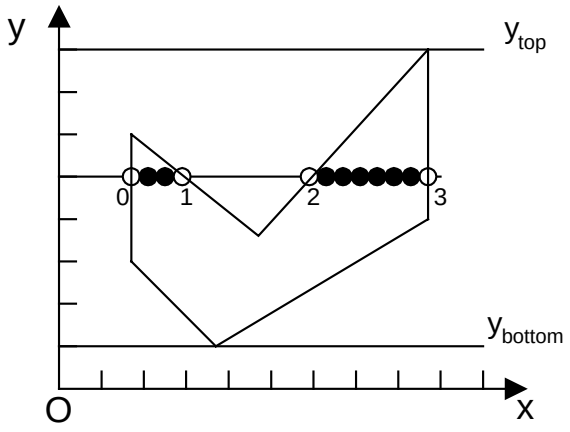
- Mot vung to thöông ñöôc xac ñònh bôi mot ñöông khep kín nao ño goi la ñöông bien. Dang ñöông bien ñôn gian thöông gap la ña giac.
- Co hai dang vung to thöông gap : to bang mot mau thuan nhat (solid fill) va to theo mot mau to (fill-pattern) nao ño.
- Viec to mau thöông ñöôc chia lam hai cong ñoan :
  - Xac ñònh vò trí cac ñiem can to mau.
  - Quyet ñònh to cac ñiem tren bang mau nao. Cong ñoan nay thöc sö phöc tap khi ta can to theo mot mau to nao ño chö khong phai to thuan mot mau.
- Co hai cach tiep can chính : to mau theo dong quet va to mau döa theo ñöông bien.
  - Phöông phap to mau döa theo dong quet se xac ñònh phan giao cua cac dong quet ke tiep nhau vôi ñöông bien cua vung to, sau ño se tien hanh to mau cac ñiem thuoc phan giao nay. Cach nay thöông ñöôc dung ñe to mau ña giac, ñöông tron, ellipse va mot so ñöông cong ñôn gian khac.
  - Phöông phap to mau döa theo ñöông bien se bat ñau tö mot ñiem ben trong vung to va tö ño loang dan ra cho ñen khi gap ñiem bien. Cach nay thöông ñöôc dung cho cac dang ñöông bien phöc tap.

Thuat toan to theo dong quet

Bai toan ñat ra : Can to mau mot ña giac cho bôi N ñænh $P_i(x_i, y_i), i = 0,...N-1$. Ña giac nay co the la ña giac loi, ña giac lom, va ca ña giac tö cat, …

Tom tat cac böôc chính cua thuat toan

- Tìm y_{top}, y_{bottom} lan löôt la gia trò lôn nhat, nho nhat cua tap cac tung ño cua cac ñænh cua ña giac ña cho: $y_{top} = \max\{y_i, (x_i, y_i) \in P\}$, $y_{bottom} = \min\{y_i, (x_i, y_i) \in P\}$.
- Öng vôi moi dong quet $y = k$, vôi k thay ñoi tö y_{bottom} ñen y_{top}, lap :
  - Tìm tat ca cac hoanh ño giao ñiem cua dong quet $y = k$ vôi cac canh cua ña giac.
  - Sap xep cac hoanh ño giao ñiem theo thö tö tang dan : $x_0, x_1, x_2, ...,$
  - To mau cac ñoan thang tren ñöông thang $y = k$ lan löôt ñöôc giôi han bôi cac cap $(x_0, x_1), (x_1, x_2), ..., (x_{2k}, x_{2k+1})$.



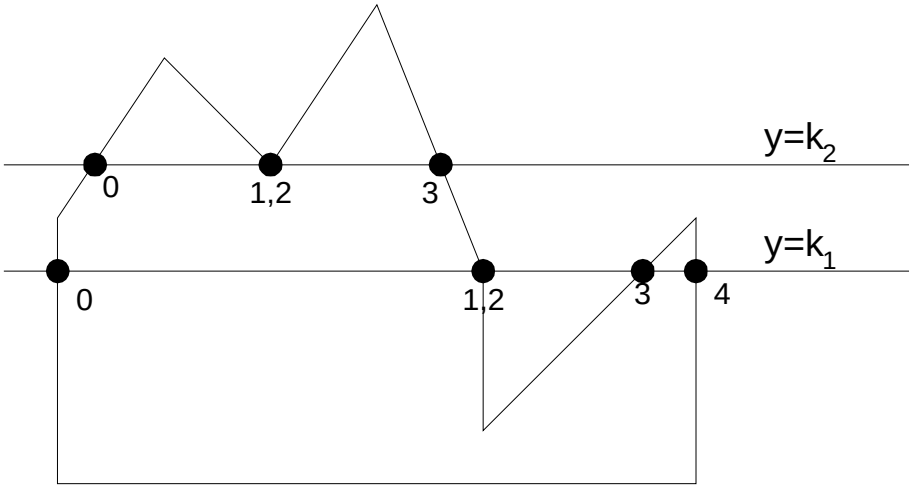
Cac van ñe ñat ra

- Han che ñöôc so canh can tìm giao ñiem öng vôi moi dong quet vì öng vôi moi dong quet, khong phai luc nao tat ca cac canh cua ña giac cung tham gia cat dong quet.

- Xac ñònh nhanh hoanh ño giao ñiem vì neu lap lai thao tac tìm giao ñiem cua canh ña giac vôi moi dong quet bang cach giai he phöông trình se ton rat nhieu thôi gian.

- Giai quyet tröông hôp so giao ñiem öng vôi tröông hôp dong quet ñi ngang qua ñænh : Neu so giao ñiem tìm ñöôc giöa cac canh ña giac va dong quet la le thì viec nhom töng cap giao ñiem ke tiep nhau ñe hình thanh cac ñoan to co the se khong chính xac. Ñieu nay chæ xay ra khi dong quet ñi ngang qua cac ñænh cua ña giac.

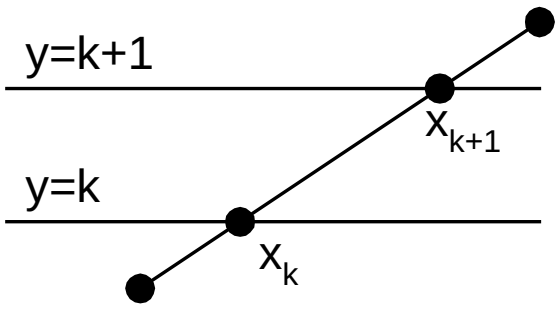
- Ngoai ra, viec tìm giao ñiem cua dong quet vôi cac canh nam ngang la mot tröông hôp ñac biet can phai co cach xö lí thích hôp



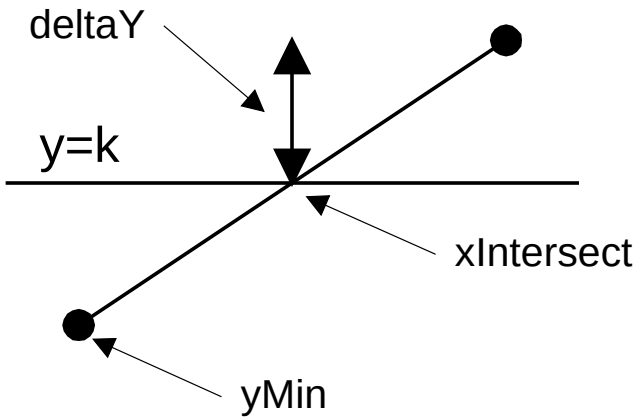
To chöc cau truc dö lieu va thuat toan

- Danh sach cac canh (Edge Table – ET) : chöa toan bo cac canh cua ña giac (***ña loai ñi cac canh nam ngang***) ñöôc sap theo thö tö tang dan cua y_{Min}.
- Danh sach cac canh kích hoat (Active Edge Table – AET) : chöa cac canh cua ña giac co the cat öng vôi dong quet hien hanh, cac canh nay ñöôc sap theo thö tö tang dan cua hoanh ño giao ñiem giöa canh va dong quet.
- Khi dong quet ñi tö bottom ñen top, cac canh thoa ñieu kien se ñöôc di chuyen tö ET sang AET:
  - Khi dong quet $y = k$ bat ñau cat mot canh, nghóa la $k \geq y_{Min}$, canh nay se ñöôc chuyen tö ET sang AET.
  - Khi dong quet khong con cat canh nay nöa, nghóa la $k > y_{Max}$, canh nay se bò loai ra khoi AET.
  - Khi khong con canh nao trong ET hay AET nöa, qua trình to mau ket thuc.
- Ñe tìm giao ñiem giöa canh ña giac va dong quet hien hanh nhanh, ta co nhan xet :

$$x_{k+1} - x_k = \frac{1}{m}\left((k+1) - k\right) = \frac{1}{m} \text{ hay } x_{k+1} = x_k + \frac{1}{m}.$$



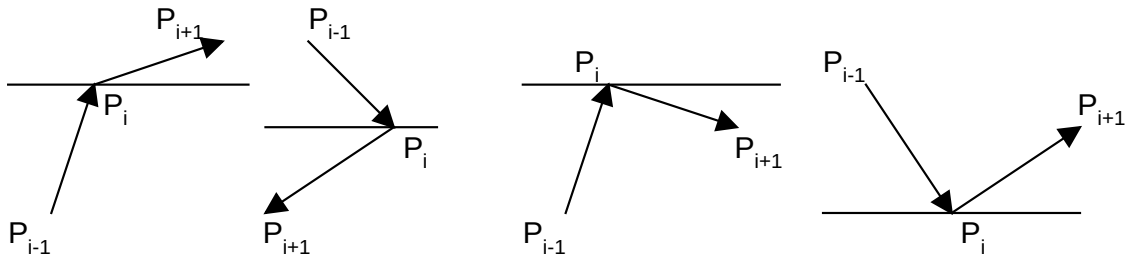
Ñe xuat cau truc dö lieu cua mot canh (EDGE)



- y_{Min}: gia trò tung ño nho nhat trong 2 ñænh cua canh.
- $xIntersect$: hoanh ño giao ñiem cua canh vôi ***dong quet hien hanh***.
- $DxPerScan$: gia trò 1/m (m la he so goc cua canh).
- $deltaY$: khoang cach tö dong quet hien hanh tôi ñænh y_{Max}. Luc nay ñieu kien $k > y_{Max}$ trô thanh $deltaY \le 0$.
- Gia trò $xIntersect$ ñöôc khôi gan ban ñau la hoanh ño cua ñænh co tung ño la y_{Min}, va gia trò $deltaY$ ñöôc khôi gan ban ñau la $y_{Max} - y_{Min} + 1$.

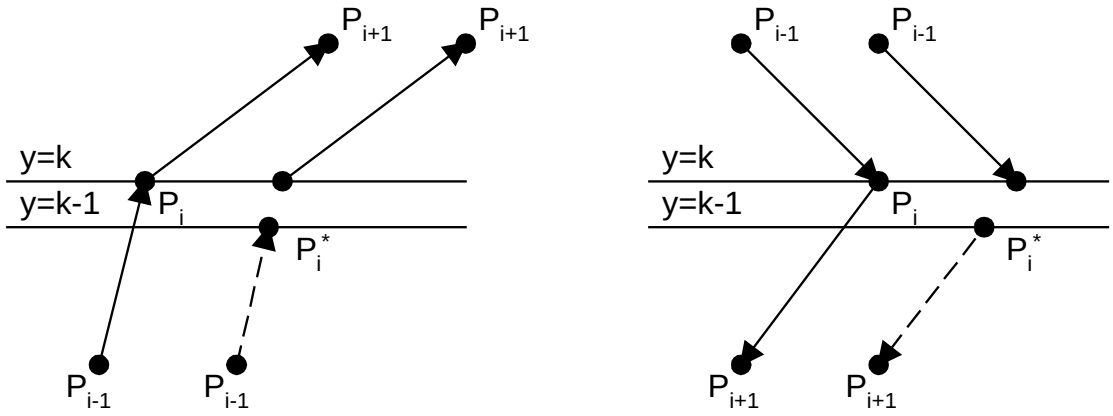
Giai quyet tröông hôp dong quet ñi qua ñænh

- Tính mot giao ñiem neu chieu cua hai canh ke cua ñænh ño co xu höông tang hay giam.
- Tính hai giao ñiem neu chieu cua hai canh ke cua ñænh ño co xu höông thay ñoi, nghóa la tang-giam hay giam-tang.

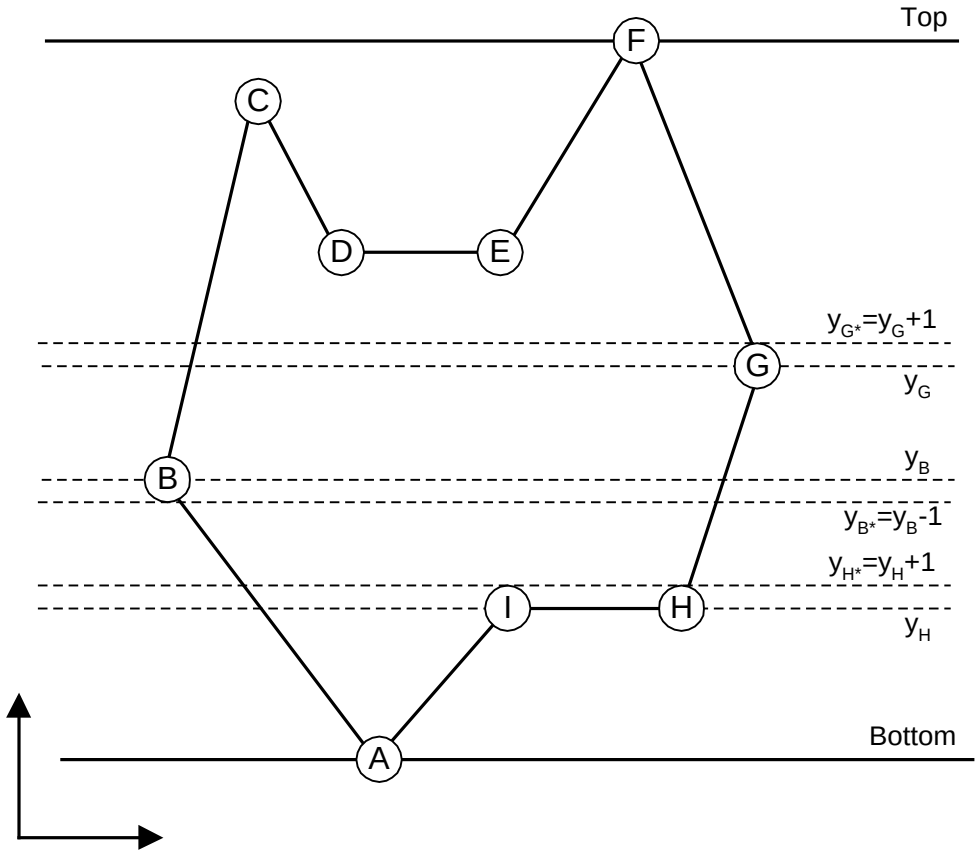


(a) (b)

- Khi cai ñat ñe khoi phai xet ñieu kien nay cho phöc tap, khi xay döng dö lieu cho moi canh tröôc khi ñöa vao ET, ngöôi ta se xö lí cac canh co ñænh tính hai giao ñiem bang cach loai ñi mot pixel tren cung cua mot trong hai canh.



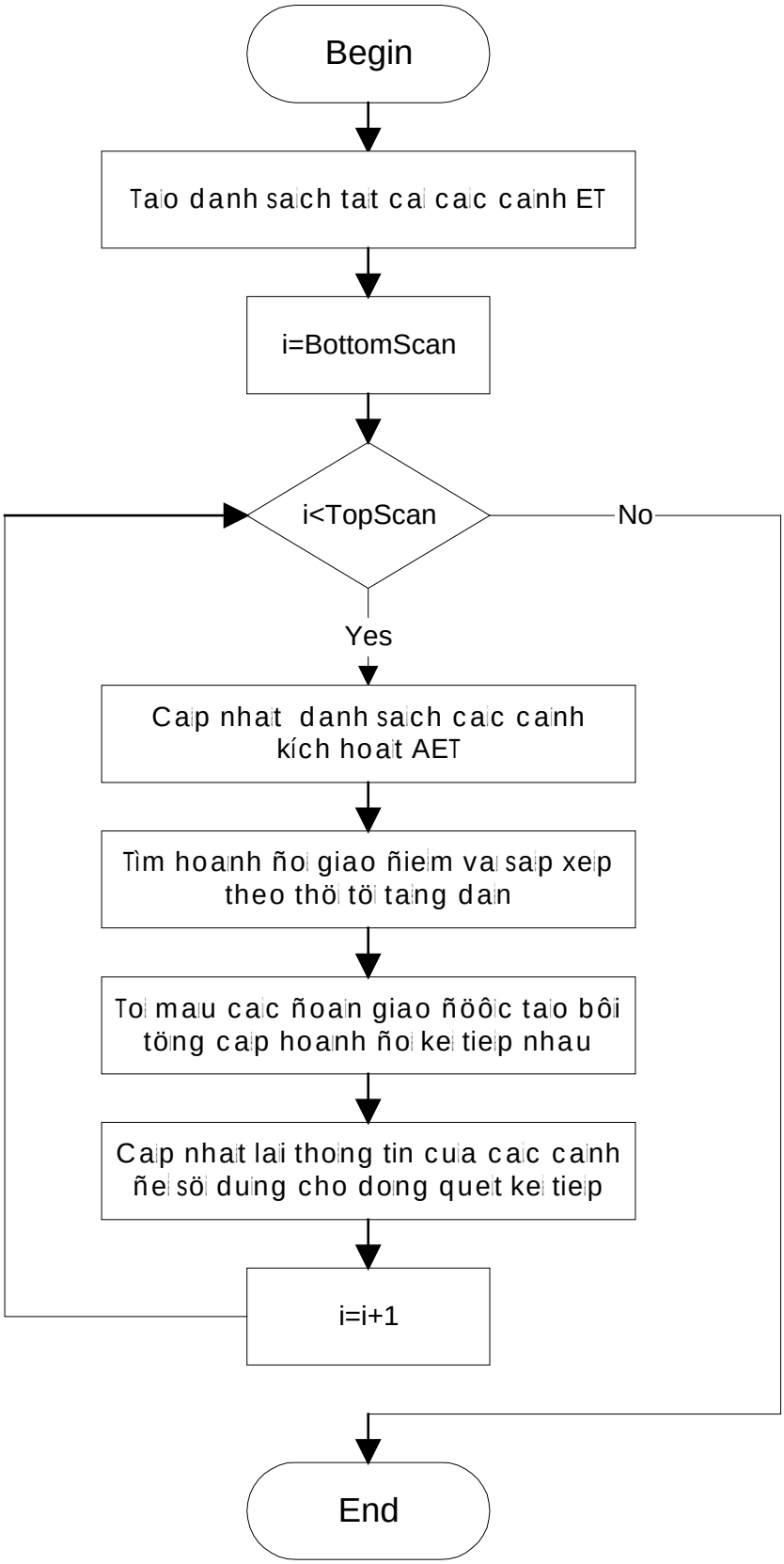
Minh hoïa thuaät toaùn



- Ban ñaàu :
  - **ET : AB*, AI, H*G, BC, G*F, DC, EF. (loaïi IH vaø DE)**
  - **AET : NULL.**
- Khi doøng queùt ñaït $y=y_A$
  - **ET : H*G, BC, G*F, DC, EF. (chuyeån AB*, AI sang AET)**
  - **AET : AB*, AI.**
- Khi doøng queùt ñaït $y=y_{H^*}$
  - **ET : BC, G*F, DC, EF. (chuyeån H*G sang AET)**
  - **AET : AB*, H*G. (loaïi AI vì khoâng coøn caét doøng queùt)**

- Khi dong quet ñaït $y=y_B$
  - **ET : G*F, DC, EF. (chuyeån BC sang AET)**
  - **AET : BC, H*G. (loaïi AB*, saép xeáp laïi H*G vaø BC)**
- Khi dong quet ñaït $y=y_{G*}$
  - **ET : DC, EF. (chuyeån G*F sang AET)**
  - **AET : BC, G*F. (loaïi H*G vì khoâng coøn caét doøng queùt)**
- Khi dong quet ñaït $y=y_D$
  - **ET : NULL. (chuyeån DC, EF sang AET)**
  - **AET : BC, DC, EF, G*F. (saép xeáp laïi BC, GF*, DC, EF)**
- Khi dong quet ñaït $y=y_C+1$
  - **ET : NULL.**
  - **AET : EF, G*F. (loaïi BC, DC vì khoâng coøn caét doøng queùt)**
- Khi dong quet ñaït $y=y_F+1$
  - **ET : NULL.**
  - **AET : NULL. (loaïi EF, G*F vì khoâng coøn caét doøng queùt).**
- Thuaät toaùn döøng taïi ñaây.

Löu ño thuat toan to mau theo dong quet



Mot so höông dan cai ñat

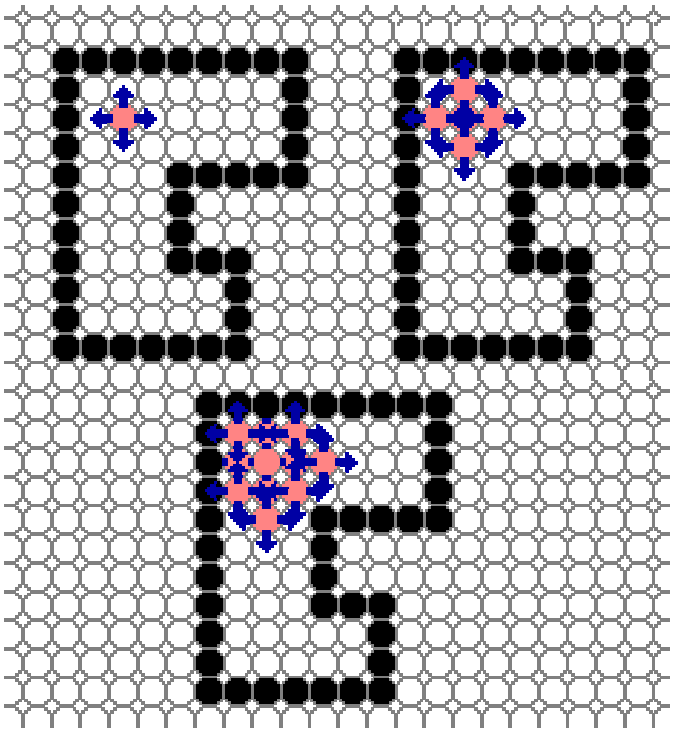
```
#define MAXVERTEX    20
#define MAXEDGE      20
#define TRUE         1
#define FALSE        0

typedef struct {
    int x;
    int y;
}POINT;

typedef struct{
    int     NumVertex;
    POINT   aVertex[MAXVERTEX];
}POLYGON;

typedef struct {
    int     NumPt;
    float   xPt[MAXEDGE];
}XINTERSECT;

typedef struct
{
    int     yMin; // Gia tri y nho nhat cua 2 dinh
    float   xIntersect; // Hoanh do giao diem cua canh & dong quet
    float   dxPerScan; // Gia tri 1/m
    int     DeltaY;
}EDGE;

typedef struct
{
    int  NumEdge;
    EDGE  aEdge[MAXEDGE];
}EDGELIST;
```

```
void PutEdgeInList(EDGELIST &EdgeList, POINT p1, POINT p2, int NextY)
{
    EDGE EdgeTmp;

    EdgeTmp.dxPerScan = float(p2.x-p1.x)/(p2.y-p1.y); // 1/m
    if(p1.y < p2.y)
    {
    /*
        Truong hop dong quet di ngang qua dinh la giao diem
        cua 2 canh co huong y cung tang
    */
        if(p2.y < NextY)
        {
            p2.y--;
            p2.x -= EdgeTmp.dxPerScan;
        }
        EdgeTmp.yMin = p1.y;
        EdgeTmp.xIntersect= p1.x;
        EdgeTmp.DeltaY = abs(p2.y-p1.y)+1;
    } // if
    else
    {
    /*
         Truong hop dong quet di ngang qua dinh la giao diem cua 2 canh co
              huong y cung giam
    */
        if(p2.y > NextY)
        {
            p2.y++;
            p2.x+= EdgeTmp.dxPerScan;
        }
        EdgeTmp.yMin = p2.y;
        EdgeTmp.xIntersect= p2.x;
        EdgeTmp.DeltaY = abs(p2.y-p1.y)+1;
    }//else
    // xac dinh vi tri chen
    int j = EdgeList.NumEdge;
    while((j>0) && (EdgeList.aEdge[j-1].yMin>EdgeTmp.yMin))
    {
        EdgeList.aEdge[j] = EdgeList.aEdge[j-1];
        j--;
    }
    // tien hanh chen dinh moi vao canh
    EdgeList.NumEdge++;
    EdgeList.aEdge[j] = EdgeTmp;
} // PutEdgeInList
```

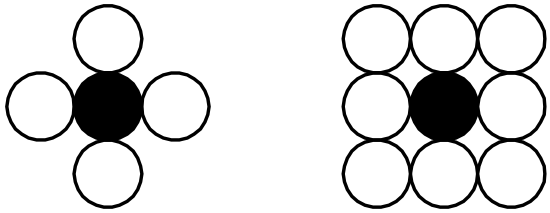
```
/*
    Tim dinh ke tiep sao cho khong nam tren cung duong thang voi dinh dang
    xet
*/
int FindNextY(POLYGON P, int id)
{
    int j = (id+1)%P.NumVertex;
    while((j<P.NumVertex)&&(P.aVertex[id].y == P.aVertex[j].y))
        j++;
    if(j<P.NumVertex)
        return (P.aVertex[j].y);
    return 0;
} // FindNextY

// Tao danh sach cac canh tu polygon da cho
void MakeSortedEdge(POLYGON P, EDGELIST &EdgeList,
                    int &TopScan, int &BottomScan)
{
    TopScan = BottomScan = P.aVertex[0].y;
    EdgeList.NumEdge = 0;
    for(int i=0; i<P.NumVertex; i++)
    {
        // Truong hop canh khong phai la canh nam ngang
        if(P.aVertex[i].y != P.aVertex[i+1].y)
            PutEdgeInList(EdgeList, P.aVertex[i], P.aVertex[i+1], FindNextY(P,
            i+1));
        // Xu li truong hop  canh nam ngang
        else
            if(P.aVertex[i+1].y > TopScan)
                TopScan = P.aVertex[i+1].y;
    }
    BottomScan = EdgeList.aEdge[0].yMin;
} //MakeSortedEdge

// Cap nhat lai hai con tro FirstId, LastId cho biet danhsach cac canh active
void UpdateActiveEdgeList(EDGELIST EdgeList, int yScan, int &FirstId, int
    &LastId)
{
    while((FirstId<EdgeList.NumEdge-1) &&(EdgeList.aEdge[FirstId].DeltaY ==
        0))
        FirstId++;
    while((LastId<EdgeList.NumEdge-1)
        &&(EdgeList.aEdge[LastId+1].yMin<=yScan))
        LastId++;
} // UpdateActiveEdgeList
```

Thuat toan to mau theo ñöông bien

- Bai toan ñat ra : Can to mau vung to neu biet ñöôc mau cua ñöông bien vung to va mot ñiem nam ben trong vung to.
- Y töông : Bat ñau tö ñiem nam ben trong vung to, kiem tra cac ñiem lan can cua no ña ñöôc to hay co phai la ñiem co mau trung mau bien hay khong, neu khong phai thì ta se to ñiem ño. Qua trình nay ñöôc lap lai cho tôi khi khong con to ñöôc nöa thì döng.



- Co hai quan ñiem ve cach to nay, ño la dung 4 ñiem lan can (hình a) hay 8 ñiem lan can (hình b).



- Cai ñat minh hoa thuat toan to mau theo ñöông bien

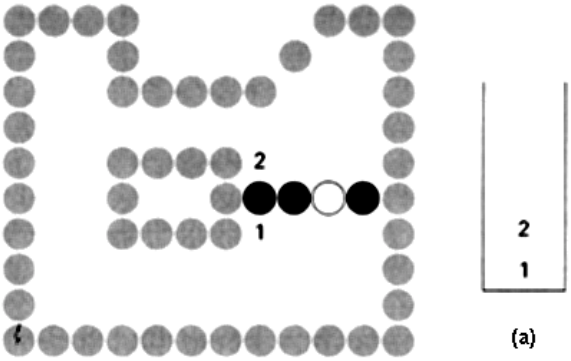
```
void BoundaryFill(int x, int y, int FillColor, int BoundaryColor)
{
    int CurrenColor;

    CurrentColor = getpixel(x,y);
    if((CurrentColor!=BoundaryColor)&&CurrentColor!= FillColor))
    {
       putpixel(x,y,FillColor);
       BoundaryFill(x-1, y, FillColor, BoundaryColor);
       BoundaryFill(x, y+1, FillColor, BoundaryColor);
       BoundaryFill(x+1, y, FillColor, BoundaryColor);
       BoundaryFill(x, y-1, FillColor, BoundaryColor);
    }
} // Boundary Fill
```

- Mot so nhan xet
  - Thuat toan co the hoat ñong khong chính xac khi co mot so ñiem nam trong vung to co mau la mau can to cua vung.
  - Viec thöc hien ñe qui lam thuat toan khong the dung cho vung to lôn.

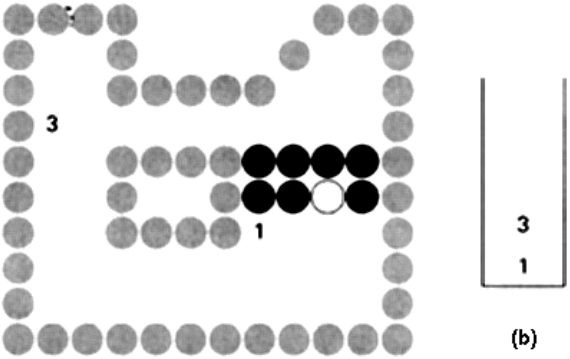
- Mot cai tien nho : nhan xet rang viec goi thöc hien ñe qui thuat toan cho 4 ñiem lan can cua ñiem hien hanh khong quan tam tôi mot trong 4 ñiem ño ña ñöôc xet ô böôc tröôc hay chöa. Ví du khi ta xet 4 ñiem lan can cua (x, y), thì khi goi thöc hien ñe qui vôi ñiem hien hanh la mot trong 4 ñiem tren, (x, y) van ñöôc xem la ñiem lan can cua chung va ñöôc goi thöc hien lai.

```
void BoundaryFillEnhanced(int x, int y, int F_Color, int B_Color)
{
    int CurrenColor;
    CurrentColor = getpixel(x,y);
    if((CurrentColor!=B_Color)&&CurrentColor!= F_Color))
    {
        putpixel(x,y,F_Color);
        FillLeft(x-1, y, F_Color, B_Color);
        FillTop(x, y+1, F_Color, B_Color);
        FillRight(x+1, y, F_Color, B_Color);
        FillBottom(x, y-1, F_Color, B_Color);
    }
} // BoundaryFillEnhanced
void FillLeft(int x, int y, int F_Color, int B_Color)
{
    int CurrenColor;
    CurrentColor = getpixel(x,y);
    if((CurrentColor!=B_Color)&&CurrentColor!= F_Color))
    {
        putpixel(x,y,F_Color);
        FillLeft(x-1, y, F_Color, B_Color);
        FillTop(x, y+1, F_Color, B_Color);
        FillBottom(x, y-1, F_Color, B_Color);
    }
} // FillLeft
```

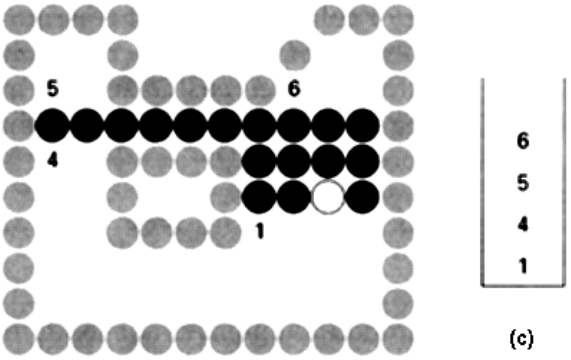
- Mot cai tien khac : khong cai ñat ñe qui ma to theo töng dong.



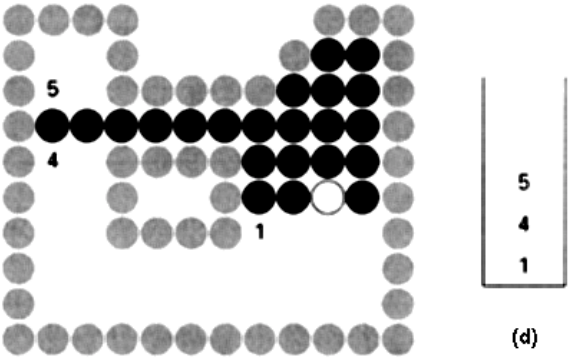
(a)



(b)



(c)



(d)