



java point

STACK IN DS

C PROGRAM TO IMPLEMENT STACK

Line Clipping:

It is performed by using the line clipping algorithm. The line clipping algorithms are:

1. Cohen Sutherland Line Clipping Algorithm
2. Midpoint Subdivision Line Clipping Algorithm
3. Liang-Barsky Line Clipping Algorithm

Cohen Sutherland Line Clipping Algorithm:

In the algorithm, first of all, it is detected whether line lies inside the screen or it is outside the screen. All lines come under any one of the following categories:

1. Visible
2. Not Visible
3. Clipping Case

1. Visible: If a line lies within the window, i.e., both endpoints of the line lies within the window. A line is visible and will be displayed as it is.

2. Not Visible: If a line lies outside the window it will be invisible and rejected. Such lines will not display. If any one of the following inequalities is satisfied, then the line is considered invisible. Let A (x_1, y_1) and B (x_2, y_2) are endpoints of line.

x_{min}, x_{max} are coordinates of the window.

y_{min}, y_{max} are also coordinates of the window.

$$x_1 > x_{max}$$

$$x_2 > x_{max}$$

$$y_1 > y_{max}$$

$$y_2 > y_{max}$$

$$x_1 < x_{min}$$

$$x_2 < x_{min}$$

$$y_1 < y_{min}$$

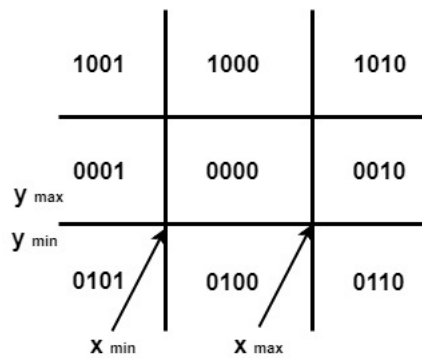
$$y_2 < y_{min}$$

3. Clipping Case: If the line is neither visible case nor invisible case. It is considered to be clipped case. First of all, the category of a line is found based on nine regions given below. All nine regions are assigned codes. Each code is of 4 bits. If both endpoints of the line have end bits zero, then the line is considered to be visible.





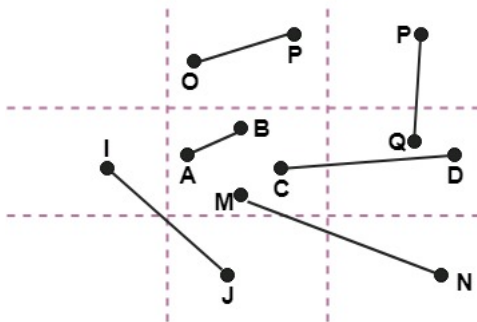
9 region



bits assigned to 9 regions

The center area is having the code, 0000, i.e., region 5 is considered a rectangle window.

Following figure show lines of various types



Line AB is the visible case

Line OP is an invisible case

Line PQ is an invisible line

Line IJ are clipping candidates

Line MN are clipping candidate

Line CD are clipping candidate

Advantage of Cohen Sutherland Line Clipping:

1. It calculates end-points very quickly and rejects and accepts lines quickly.
2. It can clip pictures much large than screen size.

Algorithm of Cohen Sutherland Line Clipping:

Step1: Calculate positions of both endpoints of the line

Step2: Perform OR operation on both of these end-points

Step3: If the OR operation gives 0000

Then

line is considered to be visible

else

Perform AND operation on both endpoints

If And \neq 0000

then the line is invisible

else

And=0000

Line is considered the clipped case.

Step4: If a line is clipped case, find an intersection with boundaries of the window

$$m = (y_2 - y_1) / (x_2 - x_1)$$

(a) If bit 1 is "1" line intersects with left boundary of rectangle window

$$y_3 = y_1 + m(x - x_1)$$

where $x = x_{wmin}$

where x_{wmin} is the minimum value of X co-ordinate of window

(b) If bit 2 is "1" line intersect with right boundary

$$y_3 = y_1 + m(x - x_1)$$

where $x = x_{wmax}$

where x_{wmax} is maximum value of X co-ordinate of the window

(c) If bit 3 is "1" line intersects with bottom boundary

$$x_3 = x_1 + (y - y_1) / m$$

where $y = y_{wmin}$

y_{wmin} is the minimum value of Y co-ordinate of the window

(d) If bit 4 is "1" line intersects with the top boundary

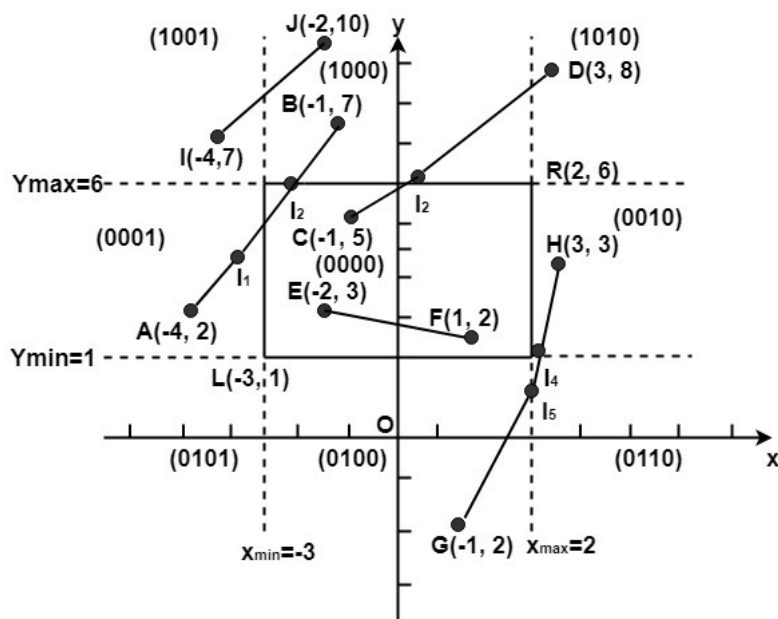
$$x_3 = x_1 + (y - y_1) / m$$

where $y = y_{wmax}$

y_{wmax} is the maximum value of Y co-ordinate of the window

Example of Cohen-Sutherland Line Clipping Algorithm:

Let R be the rectangular window whose lower left-hand corner is at L (-3, 1) and upper right-hand corner is at R (2, 6). Find the region codes for the endpoints in fig:



The region code for point (x, y) is set according to the scheme

Bit 1 = sign (y - y_{max}) = sign (y - 6) Bit 3 = sign (x - x_{max}) = sign (x - 2)

Bit 2 = sign (y_{min} - y) = sign (1 - y) Bit 4 = sign (x_{min} - x) = sign (-3 - x)

Here

$$\left\{ \begin{array}{ll} \text{sign}(a) = 1 & \text{if } a \text{ is positive} \\ 0 & \text{otherwise} \end{array} \right\}$$

So

A (-4, 2) → 0001	F (1, 2) → 0000
B (-1, 7) → 1000	G (1, -2) → 0100
C (-1, 5) → 0000	H (3, 3) → 0100
D (3, 8) → 1010	I (-4, 7) → 1001
E (-2, 3) → 0000	J (-2, 10) → 1000

We place the line segments in their appropriate categories by testing the region codes found in the problem.

Category1 (visible): \overline{EF} since the region code for both endpoints is 0000.

Category2 (not visible): \overline{IJ} since $(1001) \text{ AND } (1000) = 1000$ (which is not 0000).

Category 3 (candidate for clipping): \overline{AB} since $(0001) \text{ AND } (1000) = 0000$, \overline{CD} since $(0000) \text{ AND } (1010) = 0000$, and \overline{GH} since $(0100) \text{ AND } (0010) = 0000$.

The candidates for clipping are \overline{AB} , \overline{CD} , and \overline{GH} .

In clipping \overline{AB} , the code for A is 0001. To push the 1 to 0, we clip against the boundary line $x_{\min} = -3$. The resulting intersection point is $I_1 (-3, 3\frac{2}{3})$. We clip (do not display) $\overline{AI_1}$ and $I_1 B$. The code for I_1 is 1001. The clipping category for $\overline{I_1 B}$ is 3 since $(0000) \text{ AND } (1000) = (0000)$. Now B is outside the window (i.e., its code is 1000), so we push the 1 to a 0 by clipping against the line $y_{\max} = 6$. The resulting intersection is $I_2 (-1\frac{3}{5}, 6)$. Thus $\overline{I_2 B}$ is clipped. The code for I_2 is 0000. The remaining segment $\overline{I_1 I_2}$ is displayed since both endpoints lie in the window (i.e., their codes are 0000).

For clipping \overline{CD} , we start with D since it is outside the window. Its code is 1010. We push the first 1 to a 0 by clipping against the line $y_{\max} = 6$. The resulting intersection I_3 is $(\frac{3}{5}, 6)$, and its code is 0000. Thus $\overline{I_3 D}$ is clipped and the remaining segment $\overline{CI_3}$ has both endpoints coded 0000 and so it is displayed.

For clipping \overline{GH} , we can start with either G or H since both are outside the window. The code for G is 0100, and we push the 1 to a 0 by clipping against the line $y_{\min} = 1$. The resulting intersection point is $I_4 (2\frac{1}{5}, 1)$ and its code is 0010. We clip $\overline{GI_4}$ and work on $\overline{I_4 H}$. Segment $\overline{I_4 H}$ is not displaying since $(0010) \text{ AND } (0010) = 0010$.

Program to perform Line Clipping using Cohen Sutherland Algorithm:

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
class data
{
    int gd, gmode, x, y, xmin, ymin, ymax, xmax;
    int a1, a2;
    float x1, y1, x2, y2, x3, y3;
    int xs, ys, xe, ye;
    float maxx, maxy;
public:
    void getdata ();
    void find ();
    void clip ();
    void display (float, float, float, float);
    void checkonof (int);
    void showbit (int);
};
```



```

void data :: getdata ()
{
    cout<<"Enter the minimum and maximum coordinate of window (x, y) ";
    cin >>xmin>>ymin>>xmax>>ymax;
    cout<<"Enter the end points of the line to be clipped";
    cin >>xs>>ys>>xe>>ye;
    display (xs, ys, xe,ye);
}

```

```

void data :: display (float xs, float ys,float xe, float ye)
{
    int gd=DETECT;
    initgraph (&gd,&gmode, "");
    maxx=getmaxx();
    maxy=getmaxy();
    line (maxx/2,0,maxx/2,maxy);
    line (0, maxy/2,maxx,maxy/2);
    rectangle (maxx/2+xmin,maxy/2-ymax,maxx/2+xmax,maxy/2-ymin);
    line (maxx/2+xs,maxy/2-ys,maxx/2+xe,maxy/2-ye);
    getch();
}

```

```

void data :: find ()
{
    a1=0;
    a2=0;
    if ((ys-ymin)>0)
        a1+=8;
    if ((ymin-ys)>0)
        a1+=4;
    if ((xs-xmax)>0)
        a1+=2;
        if ((xmin-xs)>0)
            a1+=1;
    if ((ye-ymin)>0)
        a2+=8;
        if ((ymin-ye)>0)
            a2+=4;
    if ((xe-xmax)>0)
        a2+=2;
    if ((xmin-xe)>0)
        a2+=1;
    cout<<"\nThe area code of 1st point is ";
    showbit (a1);
    getch ();
    cout <<"\nThe area code of 2nd point is ";
    showbit (a2);
    getch ();
}

```

```

void data :: showbit (int n)
{
    int i,k, and;
    for (i=3;i>=0;i--)
    {
        and =1<<i;
        k = n?

```



```

    k == 0? cout << "0": cout << "1\\";
}
}
void data :: clip()
{
    int j = a1 & a2;
    if (j == 0)
    {
        cout << "\nLine is perfect candidate for clipping";
        if (a1 == 0)
        {
            else
            {
                checkonof(a1);
                x2 = x1; y2 = y1;
            }
            if (a2 == 0)
            {
                x3 = xe; y3 = ye;
            }
            else
            {
                checkonof(a2);
                x3 = x1; y3 = y1;
            }
            xs = x2; ys = y2; xe = x3; ye = y3;
            cout << endl;
            display(xs, ys, xe, ye);
            cout << "Line after clipping";
            getch();
        }
        else if ((a1 == 0) && (a2 == 0))
        {
            cout << "\n Line is in the visible region";
            getch();
        }
    }
}
void data :: checkonof(int i)
{
    int j, k, l, m;
    1 = i & 1;
    x1 = 0; y1 = 0;
    if (1 == 1)
    {
        x1 = xmin;
        y1 = ys + ((x1 - xs) / (xe - xs)) * (ye - ys);
    }
    j = i & 8;
    if (j > 0)
    {
        y1 = ymax;
        x1 = xs + (y1 - ys) / (ye - ys) * (xe - xs);
    }
    k = i & 4;

```



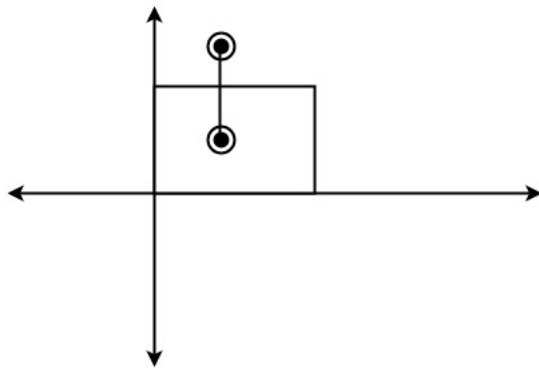
```

if (k==1)
{
    y1=ymin;
    x1=xs+((y1-ys)/(ye-ys))*(xe-xs);
}
m= i&2;
if (m==1)
{
    x1=xmax;
    y1=ys+ ((x1-xs)/(xe-xs))*(ye-ys);
}
main ()
{
    data s;
    clrscr();
    s.getdata();
    s.find();
    getch();
    closegraph ();
    return ();
}

```

Output:

The area code of 1st point is 0000
The area code of 2nd point is 1000
Line is perfect candidate for clipping


[< Prev](#)
[Next >](#)

 **For Videos Join Our Youtube Channel: [Join Now](#)**

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share



Learn Latest Tutorials



Splunk tutorial
Splunk



SPSS tutorial
SPSS



Swagger tutorial
Swagger



T-SQL tutorial
Transact-SQL



Tumblr tutorial
Tumblr



React tutorial
ReactJS



Regex tutorial
Regex



Reinforcement learning tutorial
Reinforcement Learning



R Programming tutorial
R Programming



RxJS tutorial
RxJS



React Native tutorial
React Native



Python Design Patterns
Python Design Patterns



Python Pillow tutorial
Python Pillow




Python Turtle tutorial
Python Turtle




Keras tutorial
Keras


Preparation




Aptitude
Aptitude




Logical Reasoning
Reasoning



Verbal Ability
Verbal Ability



Interview Questions
Interview Questions



Company Interview Questions
Company Questions

Trending Technologies



Artificial Intelligence
Artificial Intelligence



AWS Tutorial
AWS



Selenium tutorial
Selenium



Cloud Computing
Cloud Computing



Hadoop tutorial
Hadoop



ReactJS Tutorial
ReactJS



Data Science Tutorial
Data Science



Angular 7 Tutorial
Angular 7



Blockchain Tutorial
Blockchain



Git Tutorial
Git



Machine Learning Tutorial
Machine Learning



DevOps Tutorial
DevOps

B.Tech / MCA



DBMS tutorial
DBMS



Data Structures tutorial
Data Structures



DAA tutorial
DAA



Operating System



	Data Structures		Operating System	Computer Network	Compiler Design
 Computer Organization and Architecture Computer Organization	 Discrete Mathematics Tutorial Discrete Mathematics	 Ethical Hacking Ethical Hacking	 Computer Graphics Tutorial Computer Graphics	 Software Engineering Software Engineering	 html tutorial Web Technology
 Cyber Security tutorial Cyber Security	 Automata Tutorial Automata	 C Language tutorial C Programming	 C++ tutorial C++	 Java tutorial Java	 .Net Framework tutorial .Net
 Python tutorial Python	 List of Programs Programs	 Control Systems tutorial Control System	 Data Mining Tutorial Data Mining	 Data Warehouse Tutorial Data Warehouse	