

Lab Report

Course title: OOAD
Course code: CSE-314
3rd Year 1st Semester Examination 2022

Date of Submission: 13-08-23



Submitted to-

Dr. Mohammad Shorif Uddin
Professor
Dr. Morium Akter
Associate Professor

Sl	Class Roll	Exam Roll	Name
01	376	202188	Md. Shakhawat Hosen

Liang barsky algorithm:

Code1:

```
#include <iostream>
#include <graphics.h>

void liangBarsky(double x1, double y1, double x2, double y2,
double xmin, double ymin, double xmax, double ymax) {
    double dx = x2 - x1, dy = y2 - y1;
    double p[4] = {-dx, dx, -dy, dy};
    double q[4] = {x1 - xmin, xmax - x1, y1 - ymin, ymax - y1};
    double t_in = 0.0, t_out = 1.0;

    for (int i = 0; i < 4; ++i) {
        if (p[i] == 0 && q[i] < 0) {
            std::cout << "Line is outside of the clipping window." <<
std::endl;
            return;
        } else if (p[i] != 0) {
            double t = q[i] / p[i];
            if (p[i] < 0) {
                t_in = std::max(t_in, t);
            } else if (p[i] > 0) {
                t_out = std::min(t_out, t);
            }
        }
    }
}
```

```

    }
}
}

if (t_in > t_out) {
    std::cout << "Line is outside of the clipping window." <<
std::endl;
} else {
    double new_x1 = x1 + t_in * dx;
    double new_y1 = y1 + t_in * dy;
    double new_x2 = x1 + t_out * dx;
    double new_y2 = y1 + t_out * dy;

    line(new_x1, new_y1, new_x2, new_y2);
}
}

```

```

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    double x1 = 50, y1 = 50, x2 = 200, y2 = 200;
    double xmin = 100, ymin = 100, xmax = 300, ymax = 300;

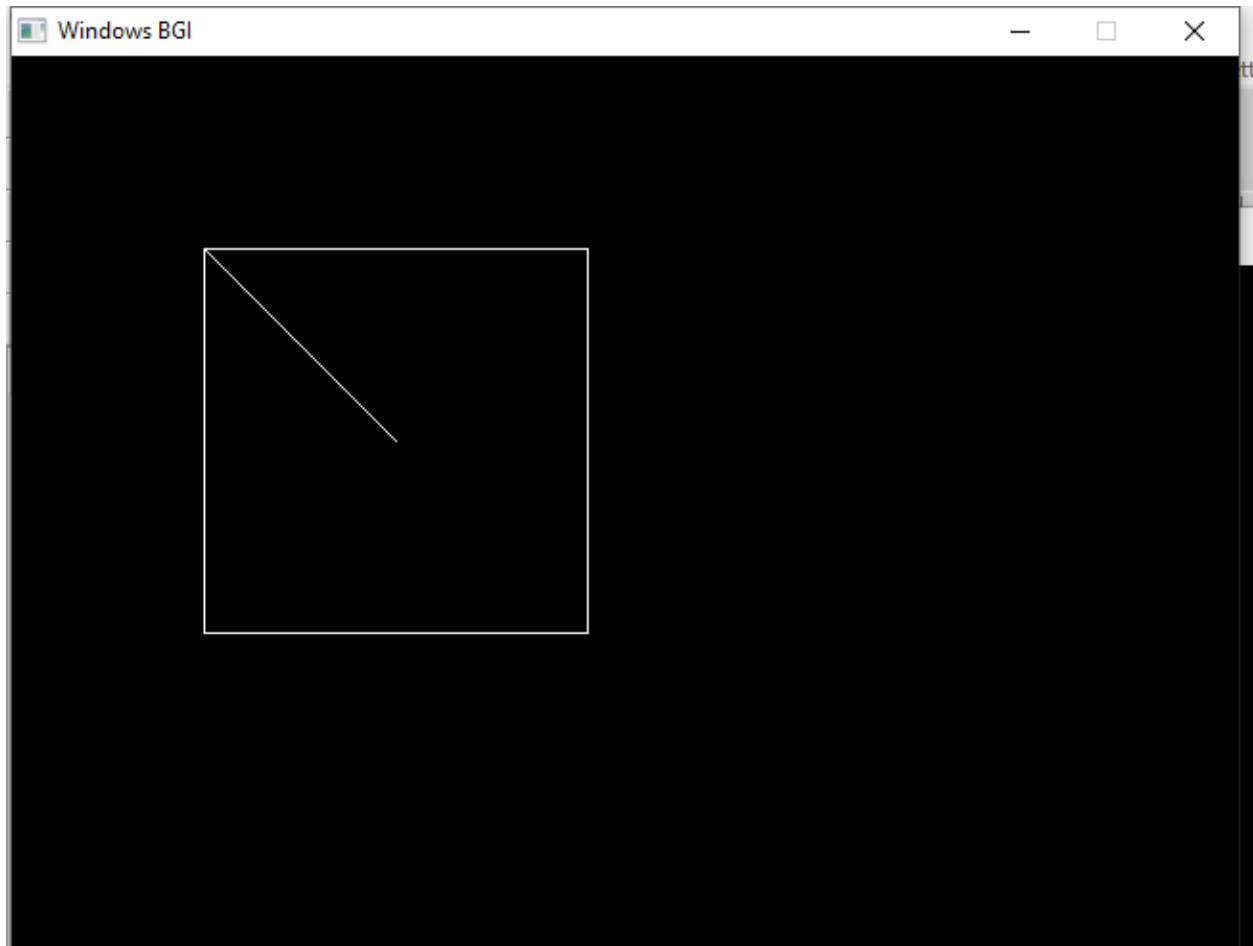
    rectangle(xmin, ymin, xmax, ymax);
    liangBarsky(x1, y1, x2, y2, xmin, ymin, xmax, ymax);

    delay(5000); // Pause for 5 seconds

    closegraph();
}

```

```
    return 0;  
}
```



Code 2:

```
#include <iostream>
#include <cmath>
#include <graphics.h>

using namespace std;

const int LEFT = 1, RIGHT = 2, BOTTOM = 4, TOP = 8;

int x_min, y_min, x_max, y_max;

int computeCode(int x, int y) {
    int code = 0;
    if (x < x_min)
        code |= LEFT;
    if (x > x_max)
        code |= RIGHT;
    if (y < y_min)
        code |= BOTTOM;
```

```

    if (y > y_max)
        code |= TOP;
    return code;
}

```

```

void liangBarsky(int x1, int y1, int x2, int y2) {
    int code1 = computeCode(x1, y1);
    int code2 = computeCode(x2, y2);
    bool accept = false;

    while (true) {
        if (!(code1 | code2)) {
            accept = true;
            break;
        } else if (code1 & code2) {
            break;
        } else {
            int codeOut = code1 ? code1 : code2;
            int x, y;

            if (codeOut & TOP) {
                x = x1 + (x2 - x1) * (y_max - y1) / (y2 - y1);
                y = y_max;
            } else if (codeOut & BOTTOM) {
                x = x1 + (x2 - x1) * (y_min - y1) / (y2 - y1);
                y = y_min;
            } else if (codeOut & RIGHT) {
                y = y1 + (y2 - y1) * (x_max - x1) / (x2 - x1);
            }
        }
    }
}

```

```

        x = x_max;
    } else if (codeOut & LEFT) {
        y = y1 + (y2 - y1) * (x_min - x1) / (x2 - x1);
        x = x_min;
    }

    if (codeOut == code1) {
        x1 = x;
        y1 = y;
        code1 = computeCode(x1, y1);
    } else {
        x2 = x;
        y2 = y;
        code2 = computeCode(x2, y2);
    }
}

}

if (accept) {
    line(x1, y1, x2, y2);
}

}

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

```

```
    cout << "Enter the coordinates of the clipping window  
(x_min y_min x_max y_max): ";  
    cin >> x_min >> y_min >> x_max >> y_max;  
  
    rectangle(x_min, y_min, x_max, y_max);  
  
    int x1, y1, x2, y2;  
    cout << "Enter the endpoints of the line (x1 y1 x2 y2): ";  
    cin >> x1 >> y1 >> x2 >> y2;  
  
    line(x1, y1, x2, y2);  
  
    delay(10000);  
    closegraph();  
    return 0;  
}
```


Output:

Code3:

```
#include <iostream>
#include <cmath>
#include <graphics.h>

using namespace std;

const int LEFT = 1, RIGHT = 2, BOTTOM = 4, TOP = 8;

int x_min, y_min, x_max, y_max;

int computeCode(int x, int y) {
    int code = 0;
    if (x < x_min)
        code |= LEFT;
```

```

    if (x > x_max)
        code |= RIGHT;
    if (y < y_min)
        code |= BOTTOM;
    if (y > y_max)
        code |= TOP;
    return code;
}

```

```

void liangBarsky(int x1, int y1, int x2, int y2) {
    int code1 = computeCode(x1, y1);
    int code2 = computeCode(x2, y2);
    bool accept = false;

    while (true) {
        if (!(code1 | code2)) {
            accept = true;
            break;
        } else if (code1 & code2) {
            break;
        } else {
            int codeOut = code1 ? code1 : code2;
            int x, y;

            if (codeOut & TOP) {
                x = x1 + (x2 - x1) * (y_max - y1) / (y2 - y1);
                y = y_max;
            } else if (codeOut & BOTTOM) {
                x = x1 + (x2 - x1) * (y_min - y1) / (y2 - y1);
                y = y_min;
            }
        }
    }
}

```

```

    } else if (codeOut & RIGHT) {
        y = y1 + (y2 - y1) * (x_max - x1) / (x2 - x1);
        x = x_max;
    } else if (codeOut & LEFT) {
        y = y1 + (y2 - y1) * (x_min - x1) / (x2 - x1);
        x = x_min;
    }

    if (codeOut == code1) {
        x1 = x;
        y1 = y;
        code1 = computeCode(x1, y1);
    } else {
        x2 = x;
        y2 = y;
        code2 = computeCode(x2, y2);
    }
}

if (accept) {
    line(x1, y1, x2, y2);
}

}

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

```

```

    cout << "Enter the coordinates of the clipping window (x_min
y_min x_max y_max): ";
    cin >> x_min >> y_min >> x_max >> y_max;

    rectangle(x_min, y_min, x_max, y_max);

    int numLines;
    cout << "Enter the number of lines to clip: ";
    cin >> numLines;

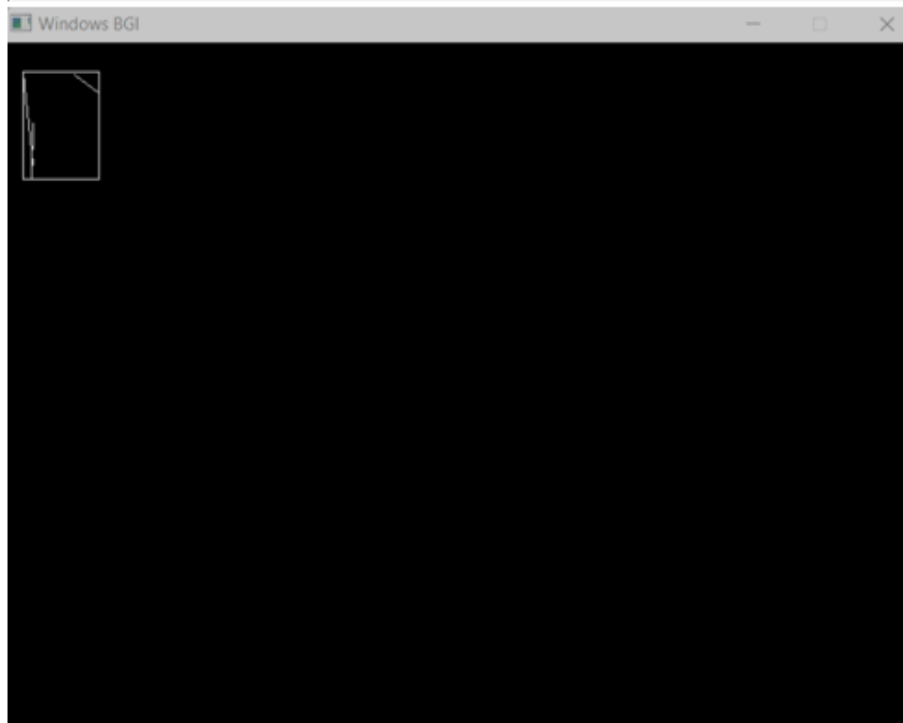
    for (int i = 0; i < numLines; i++) {
        int x1, y1, x2, y2;
        cout << "Enter endpoints of line " << (i + 1) << " (x1 y1 x2
y2): ";
        cin >> x1 >> y1 >> x2 >> y2;
        liangBarsky(x1, y1, x2, y2);
    }

    delay(10000);
    closegraph();
    return 0;
}

```

Output:

```
F:\395\Lab4\Lam.exe
Enter the coordinates of the clipping window (x_min y_min x_max y_max): 12 20 65 95
Enter the number of lines to clip: 3
Enter endpoints of line 1 (x1 y1 x2 y2): 12 13 19 85
Enter endpoints of line 2 (x1 y1 x2 y2): 19 56 18 95
Enter endpoints of line 3 (x1 y1 x2 y2): 65 35 48 22
```



Windows BGI

