TRIBHUVAN UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu
2022

Lab Report: III

"Implementation of Segment Intersection"

Submitted By:
Sudhan Kandel
Semester: 2nd
Roll no: 2

Submitted To:
Asst.Prof. Jagdish Bhatta

1. Write program for proper and improper intersection

```
import matplotlib.pyplot as plt
class LineIntersect:
    def __init__(self):
        self.points=[]
        for i,j in enumerate(['Line 1','Line 2']):
            for k in ['starting','ending']:
                x,y=input("Please Enter "+j+"'s "+k+" point's X-Cordinate and Y-
cordinate....").split(',')
                x=float(x)
                y=float(y)
                if len(self.points)==i:
                    self.points.append([[x,y]])
                else:
                    self.points[i].append([x,y])
    def check_intersect(self,a,b,c,d):
        abc=((b[0]-a[0])*(c[1]-a[1]))-((c[0]-a[0])*(b[1]-a[1]))
        abd=((b[0]-a[0])*(d[1]-a[1]))-((d[0]-a[0])*(b[1]-a[1]))
        if (abd<0 and abc>0) or (abd>0 and abc<0):
            return 1
        else:
            return 0
    def slope(self,x1,x2,x3,y1,y2,y3):
        c = x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)
        return c
    def colinear(self,a ,b,c):
        ac=slope(a[0],b[0],c[0],a[1],b[1],c[1])
        if ac==0:
            return 1
        else:
```

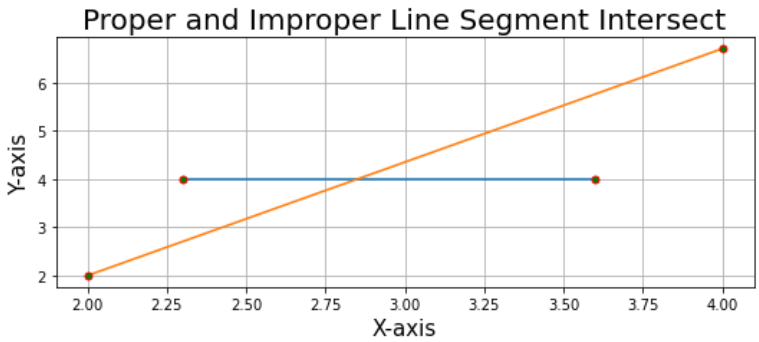
```
        return 0
    def type_intersect(self):
        l1=self.points[0]
        l2=self.points[1]
        a=l1[0]
        b=l1[1]
        c=l2[0]
        d=l2[1]
        result=self.check_intersect(a,b,c,d)
        if result==1:
            return ("Proper Intersection")
        else:
            if self.colinear(a,b,c)==1:
                if a[1]<=c[1]<=b[1] or a[1]>=c[1]>=b[1]:
                    return ("improper Intersection")
                else:
                    return ("Segment are not interset")
            else:
                return ("Line are not interset")

    def displayinfo(self):
        print("\n.....")
        print("Name: Sudhan Kandel","\nRoll No: 2","\nSection: A")
        print("\n.....")
        print("Datastructure of Segments is: ",self.points)
        print("\n.....")
        print("Type of intersect for the given line segment is: \n", self.type_intersect())
        print("\n.....")
        self.visualization()
```

```
    def visualization(self):
        plt.rcParams["figure.figsize"] = [7.50, 3.50]
        plt.rcParams["figure.autolayout"] = True
        l1=self.points[0]
        l2=self.points[1]
        plt.grid()
        plt.plot([l1[0][0],l1[1][0]],[l1[0][1],l1[1][1]], linestyle="-", marker="o",
markersize=5, markeredgecolor="red", markerfacecolor="green")
        plt.plot([l2[0][0],l2[1][0]],[l2[0][1],l2[1][1]], linestyle="-", marker="o",
markersize=5, markeredgecolor="red", markerfacecolor="green")
        plt.title("Proper and Improper Line Segment Intersect",fontdict={'fontsize':20})
        plt.xlabel("X-axis",fontdict={'fontsize':15})
        plt.ylabel("Y-axis",fontdict={'fontsize':15})
        plt.savefig('line.png')
        plt.show()
intersect=LineIntersect()
intersect.displayinfo()
```

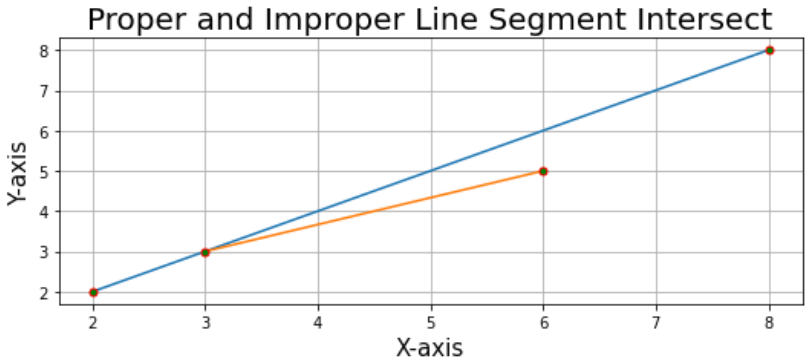
OUTPUT:

```
Please Enter Line 1's starting point's X-Cordinate and Y-cordinate....2
.3,4
Please Enter Line 1's ending point's X-Cordinate and Y-cordinate....3.6
,4
Please Enter Line 2's starting point's X-Cordinate and Y-cordinate....2
,2
Please Enter Line 2's ending point's X-Cordinate and Y-cordinate....4,6
.7

................................
Name: Sudhan Kandel
Roll No: 2
Section: A

................................
Datastructure of Segments is:  [[[2.3, 4.0], [3.6, 4.0]], [[2.0, 2.0],
[4.0, 6.7]]]

................................
Type of intersect for the given line segment is:
```

```
 Proper Intersection

.................................
```




```
Please Enter Line 1's starting point's X-Cordinate and Y-cordinate....2
,2
Please Enter Line 1's ending point's X-Cordinate and Y-cordinate....8,8
Please Enter Line 2's starting point's X-Cordinate and Y-cordinate....3
,3
Please Enter Line 2's ending point's X-Cordinate and Y-cordinate....6,5

.................................
Name: Sudhan Kandel
Roll No: 2
Section: A

.................................
Datastructure of Segments is:  [[[2.0, 2.0], [8.0, 8.0]], [[3.0, 3.0],
[6.0, 5.0]]]

.................................
Type of intersect for the given line segment is:
 improper Intersection

.................................
```



Code link:

https://github.com/sudhankandel/CGlab/blob/main/SUDHAN_KANDEL_2_Lab3.ipynb