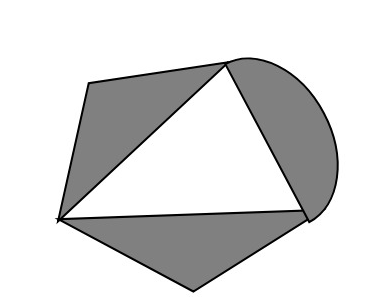
Flood Fill Algorithm:

In this method, a point or seed which is inside region is selected. This point is called a seed point. Then four connected approaches or eight connected approaches is used to fill with specified color.

The flood fill algorithm has many characters similar to boundary fill. But this method is more suitable for filling multiple colors boundary. When boundary is of many colors and interior is to be filled with one color we use this algorithm.



In fill algorithm, we start from a specified interior point (x, y) and reassign all pixel values are currently set to a given interior color with the desired color. Using either a 4-connected or 8-connected approaches, we then step through pixel positions until all interior points have been repainted.

Disadvantage:

1. Very slow algorithm
2. May be fail for large polygons
3. Initial pixel required more knowledge about surrounding pixels.

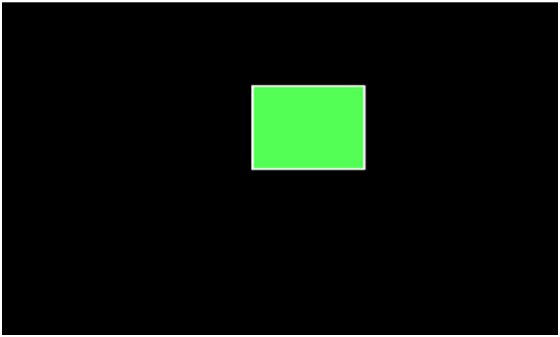
Algorithm:

1. Procedure floodfill (x, y,fill\_ color, old\_color: integer)
2. If (getpixel (x, y)=old\_color)
3. {
4. setpixel (x, y, fill\_color);
5. fill (x+1, y, fill\_color, old\_color);
6. fill (x-1, y, fill\_color, old\_color);
7. fill (x, y+1, fill\_color, old\_color);
8. fill (x, y-1, fill\_color, old\_color);
9. }
10. }

Program1: To implement 4-connected flood fill algorithm:

1. #include<stdio.h>
2. #include<conio.h>
3. #include<graphics.h>
4. #include<dos.h>
5. **void** flood(**int**,**int**,**int**,**int**);
6. **void** main()
7. {
8. intgd=DETECT,gm;
9. initgraph(&gd,&gm,"C:/TURBOC3/bgi");
10. rectangle(50,50,250,250);
11. flood(55,55,10,0);
12. getch();
13. }
14. **void** flood(intx,inty,intfillColor, intdefaultColor)
15. {
16. **if**(getpixel(x,y)==defaultColor)
17. {
18. delay(1);
19. putpixel(x,y,fillColor);
20. flood(x+1,y,fillColor,defaultColor);
21. flood(x-1,y,fillColor,defaultColor);
22. flood(x,y+1,fillColor,defaultColor);
23. flood(x,y-1,fillColor,defaultColor);
24. }
25. }

**Output:**



Program2: To implement 8-connected flood fill algorithm:

1. #include<stdio.h>
2. #include<graphics.h>
3. #include<dos.h>
4. #include<conio.h>
5. **void** floodfill(intx,inty,intold,intnewcol)
6. {
7. **int** current;
8. current=getpixel(x,y);
9. **if**(current==old)
10. {
11. delay(5);
12. putpixel(x,y,newcol);
13. floodfill(x+1,y,old,newcol);
14. floodfill(x-1,y,old,newcol);
15. floodfill(x,y+1,old,newcol);
16. floodfill(x,y-1,old,newcol);
17. floodfill(x+1,y+1,old,newcol);
18. floodfill(x-1,y+1,old,newcol);
19. floodfill(x+1,y-1,old,newcol);
20. floodfill(x-1,y-1,old,newcol);
21. }
22. }
23. **void** main()
24. {
25. intgd=DETECT,gm;
26. initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
27. rectangle(50,50,150,150);
28. floodfill(70,70,0,15);
29. getch();
30. closegraph();
31. }

**Output:**

