

SANEER ARORA

BCA 6 B

41

Computer Graphics

1- WAP to Floodfill Algorithm:-

```
#include <stdio.h>
#include <graphics.h>
#include <dos.h>
#include <conio.h>

void Floodfill (int x, int y, int old, int newcol)
{
    int current;
    current = getpixel(x, y);
    if (current == old)
    {
        delay(5);
        putpixel(x, y, newcol);
        floodfill(x+1, y, old, newcol);
        floodfill(x-1, y, old, newcol);
        floodfill(x, y+1, old, newcol);
        floodfill(x, y-1, old, newcol);
        floodfill(x+1, y+1, old, newcol);
        floodfill(x-1, y+1, old, newcol);
    }
}
```

```
floodfill (u+1, v-1, old, newcol);  
floodfill (u-1, v-1, old, newcol);  
}
```

```
}
```

```
void main()
```

```
{
```

```
int gd = DETECT, gm;  
initgraph (&ga, &gm, "");  
rectangle (50, 50, 150, 150);  
floodfill (70, 70, 0, 15);  
getch();  
closegraph();
```

```
}
```


ALGORITHM

Step 1 - Initialize the value of seed points $(seedx, seedy, fcolor, and dcol)$.

Step 2 - Define the boundary values of the polygon

Step 3 - Check if the current seed point is of default color then repeat the steps 4 and 5 till the boundary pixels ~~reached~~ reached

Step 4 - Change the default colour with the fill color at the seed point.
Set pixel $(seedx, seedy, fcol)$

Step 5 - Recursively follow the procedure with four neighbourhood points

FloodFill($seedx-1, seedy, fcol, dcol$)

FloodFill($seedx+1, seedy, fcol, dcol$)

FloodFill($seedx, seedy-1, fcol, dcol$)

FloodFill($seedx, seedy+1, fcol, dcol$)

FloodFill($seedx-1, seedy+1, fcol, dcol$)

FloodFill($seedx+1, seedy+1, fcol, dcol$)

FloodFill($seedx+1, seedy-1, fcol, dcol$)

FloodFill($seedx-1, seedy-1, fcol, dcol$)

~~FloodFill~~

Step 6 - Exit

