

# Connected Components

```
In [2]: import cv2
import numpy as np
import matplotlib.pyplot as plt

In [3]: shp = (25,25)
img = np.floor(np.random.random(shp) + 0.5)

In [4]: def four_way(out, i, j, color, label):

    if i < 0 or i >= shp[0]:
        return
    if j < 0 or j >= shp[1]:
        return
    if vis[i][j] or img[i][j]==1:
        return

    vis[i][j] = True
    out[i][j] = color
    labels[i][j] = label
    four_way(out, i - 1, j, color, label)
    four_way(out, i + 1, j, color, label)
    four_way(out, i, j - 1, color, label)
    four_way(out, i, j + 1, color, label)

In [5]: def eight_way(out, i, j, color, label):
    if i < 0 or i >= shp[0]:
        return
    if j < 0 or j >= shp[1]:
        return
    if vis[i][j] or img[i][j] == 1:
        return

    vis[i][j] = True
    out[i][j] = color
    labels[i][j] = label

    eight_way(out, i - 1, j, color, label)
    eight_way(out, i + 1, j, color, label)
```

```

eight_way(out, i, j - 1, color,label)
eight_way(out, i, j + 1, color,label)
eight_way(out, i - 1, j - 1, color,label)
eight_way(out, i - 1, j + 1, color,label)
eight_way(out, i + 1, j - 1, color,label)
eight_way(out, i + 1, j + 1, color,label)

```

In [6]: `def m_conn(x1,y1,x2,y2):`

```

four_way_n_p1 = {}
four_way_n_p1[(x1-1,y1-1)] = img[x1-1][y1-1]
four_way_n_p1[(x1-1,y1)] = img[x1-1][y1]
four_way_n_p1[(x1,y1-1)] = img[x1][y1-1]
four_way_n_p1[(x1,y1)] = img[x1][y1]

n_p2 = [(x2-1,y2-1),(x2-1,y2),(x2,y2-1),(x2,y2)]

for cord in n_p2:
    if cord in four_way_n_p1.keys():
        if img[cord[0]][cord[1]] == 1:
            return False

return True

```

In [7]: `def m_way(out, i, j, color,label):`

```

if i<0 or i>= shp[0]:
    return
if j<0 or j>= shp[1]:
    return
if vis[i][j] or img[i][j] == 1:
    return

vis[i][j] = True
out[i][j] = color
labels[i][j] = label

m_way(out, i - 1, j, color,label)
m_way(out, i + 1, j, color,label)
m_way(out, i, j - 1, color,label)
m_way(out, i, j + 1, color,label)

if m_conn(i,j,i-1,j-1):
    m_way(out, i - 1, j - 1, color,label)
elif m_conn(i,j,i-1,j+1):
    m_way(out, i - 1, j + 1, color,label)
elif m_conn(i,j,i+1,j-1):
    m_way(out, i + 1, j - 1, color,label)
elif m_conn(i,j,i+1,j+1):
    m_way(out, i + 1, j + 1, color,label)

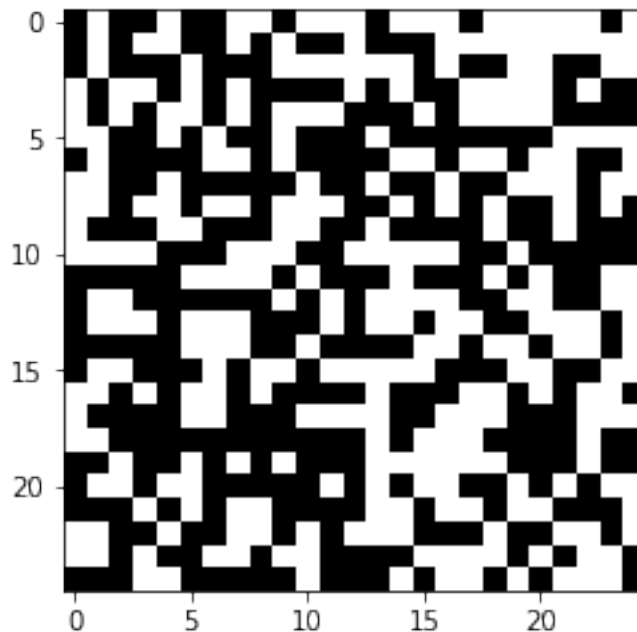
```

```
In [8]: def plot(out):
        fig, ax = plt.subplots(figsize=(20,10))

        ax.imshow(out)
        for i in range(shp[0]):
            for j in range(shp[1]):
                c = labels[j][i]
                ax.text(i, j, str(c), va='center', ha='center')
```

```
In [9]: fig = plt.figure(100)
        fig.canvas.set_window_title('Main')
        plt.imshow(img, cmap="Greys")
```

```
Out[9]: <matplotlib.image.AxesImage at 0x1254fed30>
```



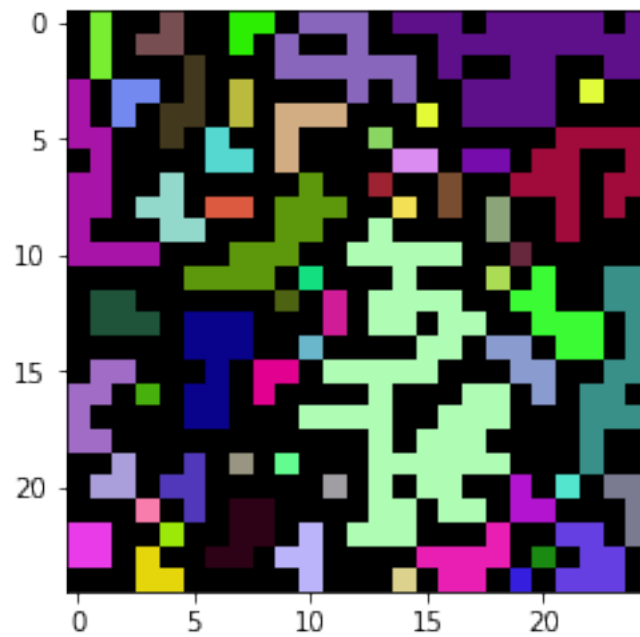
```
In [10]: vis = np.zeros(shp,dtype=bool)
        out = np.zeros(shp + (3, ), dtype=int)
        labels = np.zeros(shp,dtype=int)
        label=1
        for i in range(shp[0]):
            for j in range(shp[1]):
                if vis[i][j] or img[i][j]==1:
                    continue

                color = np.random.randint(0, 255, 3)
                four_way(out, i, j, color,label)
```

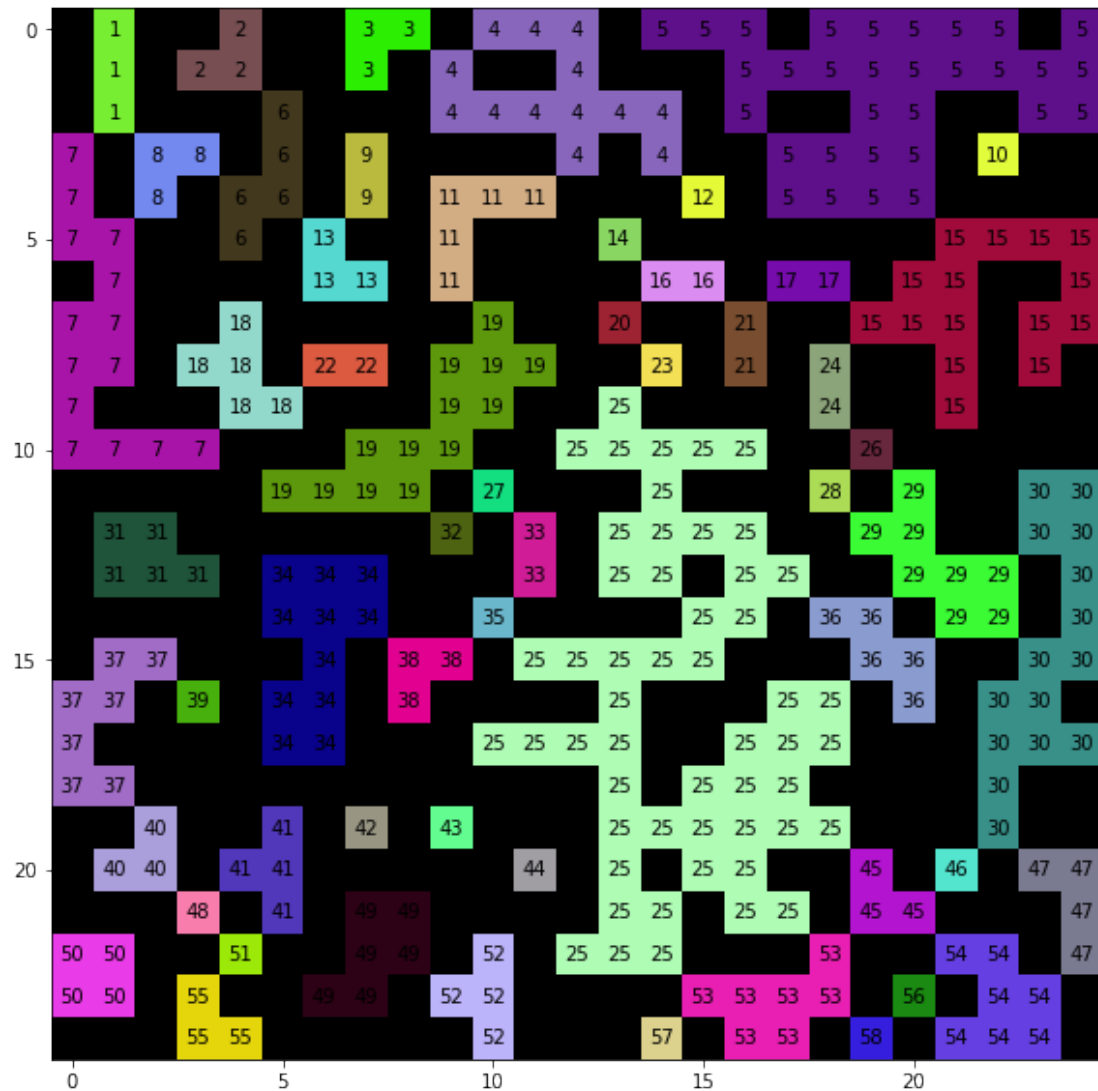
```
label+=1
```

```
fig = plt.figure(200)  
fig.canvas.set_window_title('4-Way')  
plt.imshow(out)
```

```
Out[10]: <matplotlib.image.AxesImage at 0x12566d748>
```



```
In [11]: plot(out)
```



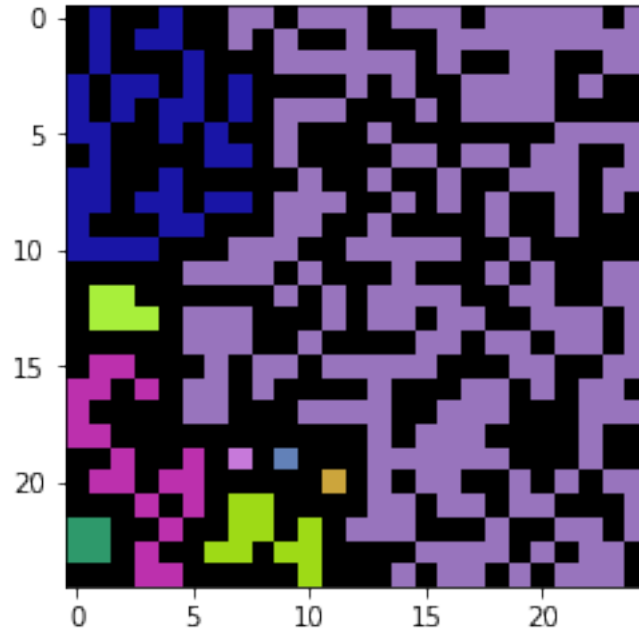
```
In [12]: vis = np.zeros(shp, dtype=bool)
out = np.zeros(shp + (3, ), dtype=int)
label=1
for i in range(shp[0]):
    for j in range(shp[1]):

        if vis[i][j] or img[i][j]==1:
            continue
        color = np.random.randint(0,255,3)
        eight_way(out, i, j, color,label)
        label+=1

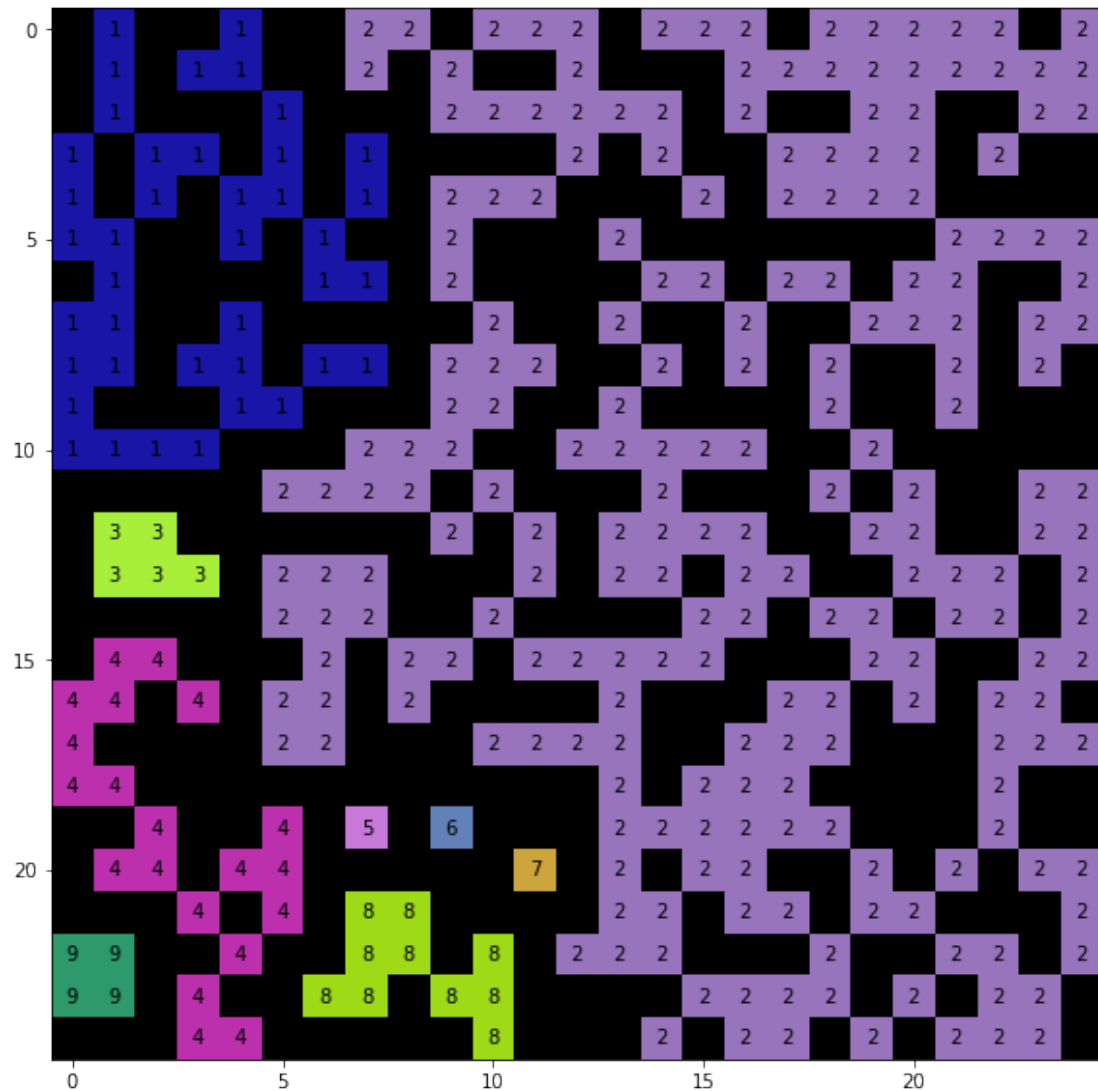
fig = plt.figure(300)
```

```
fig.canvas.set_window_title('8-Way')  
plt.imshow(out)
```

Out[12]: <matplotlib.image.AxesImage at 0x125f7dd30>



In [13]: plot(out)

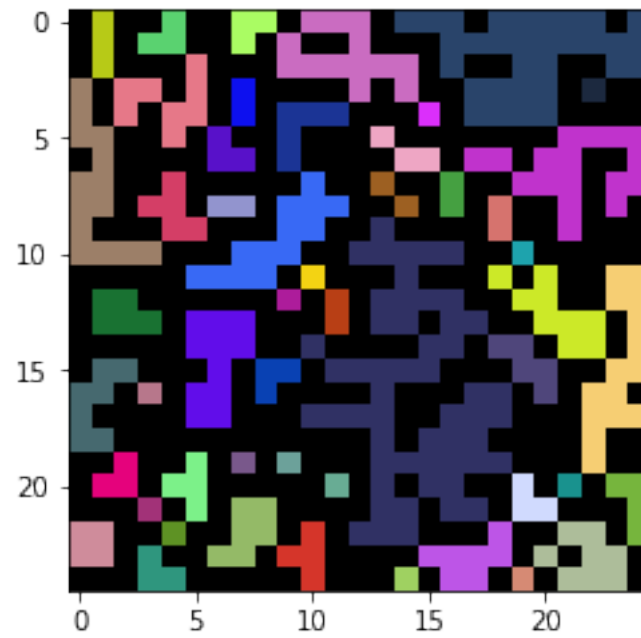


```
In [14]: vis = np.zeros(shp, dtype=bool)
out = np.zeros(shp + (3, ), dtype=int)
label=1
for i in range(shp[0]):
    for j in range(shp[1]):
        if vis[i][j] or img[i][j]==1:
            continue

        color = np.random.randint(0, 255, 3)
        m_way(out, i, j, color,label)
        label+=1

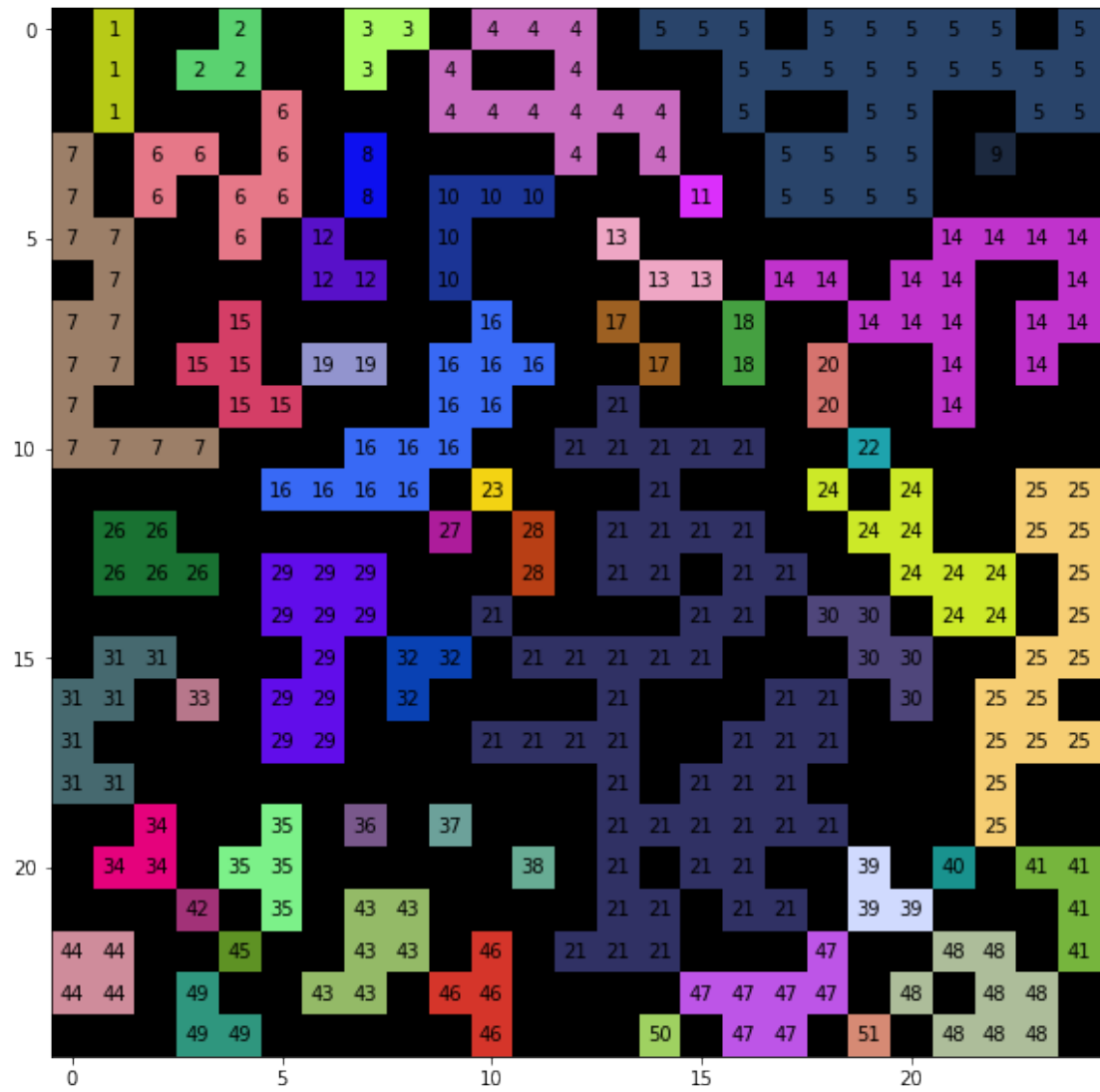
fig = plt.figure(400)
```

```
fig.canvas.set_window_title('m-Way')  
plt.imshow(out)  
plt.show()
```



```
In [15]: plot(out)
```





In [ ]: