

Connected Component Analysis

Minyoung Chung

Computer Graphics and Image Processing Laboratory

Seoul National University



Aug 6, 2014



Contents

- Connected Component Labeling
 - Implementation

What is Connected-Component Labeling ?

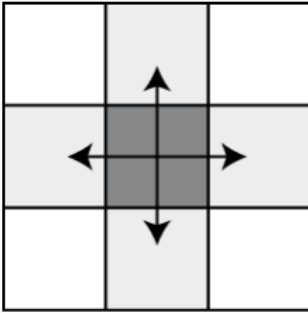
- Detection of connected objects in an image.
- Subsets of connected components are uniquely labeled.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

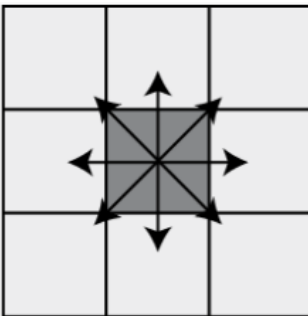
0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	2	2	2	0
0	1	0	0	2	0	0
0	0	0	0	0	0	0

Connected-Component Labeling

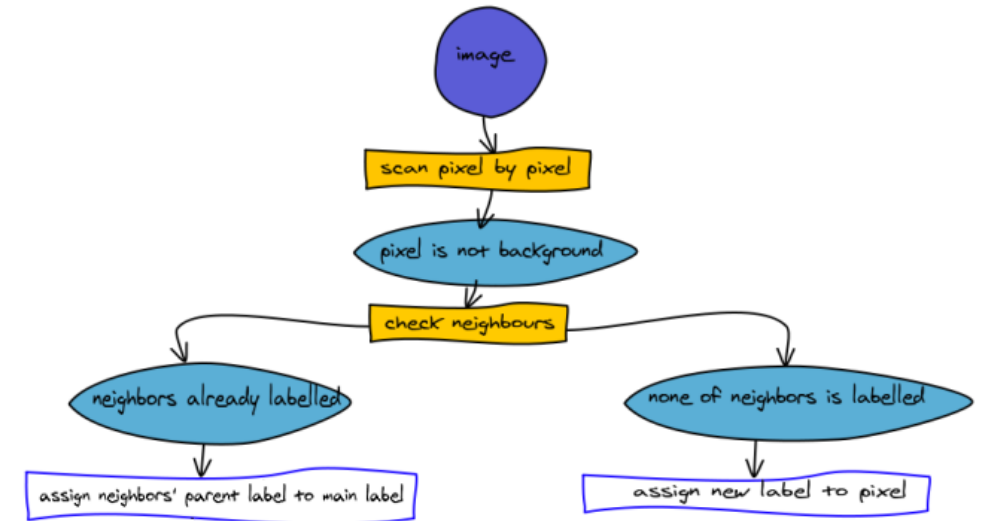
- Connectivity
 - 4-connectivity



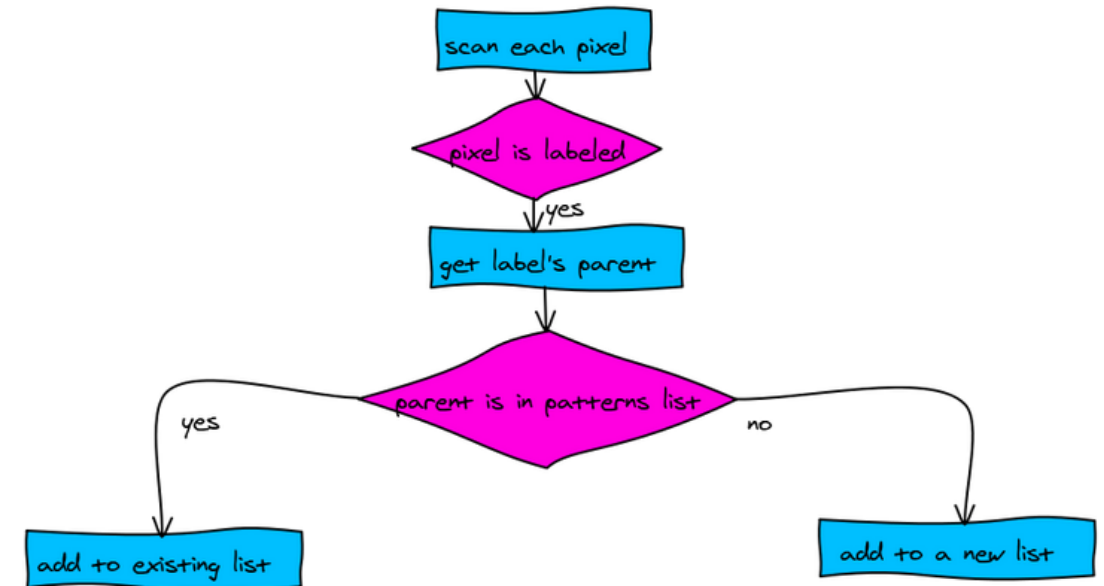
- 8-connectivity



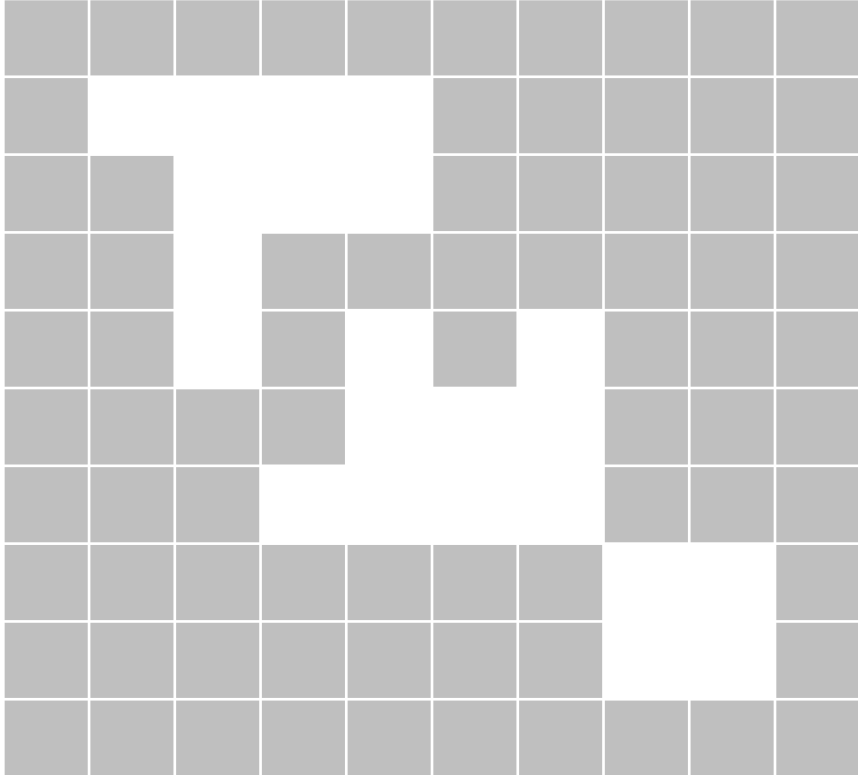
First pass (assigning labels)



Second pass (aggregation)

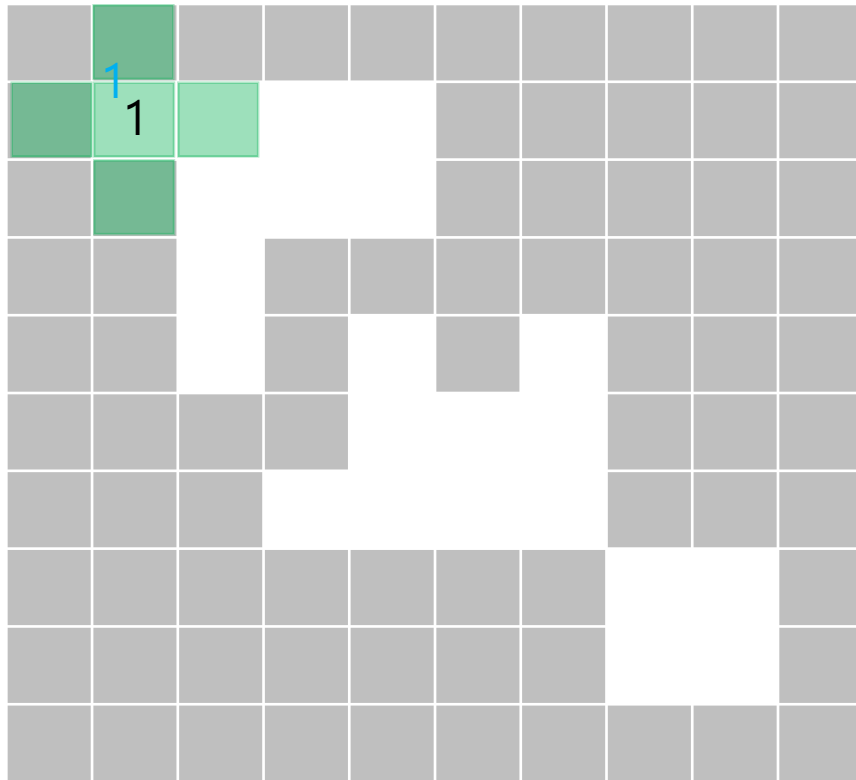


Algorithm for CCL (Connected-Component Labeling)



- Binary Input Image.
 - "White" for *foreground*
 - "Gray" for *background*
- We are not interested in background region.

Algorithm for CCL (Connected-Component Labeling)



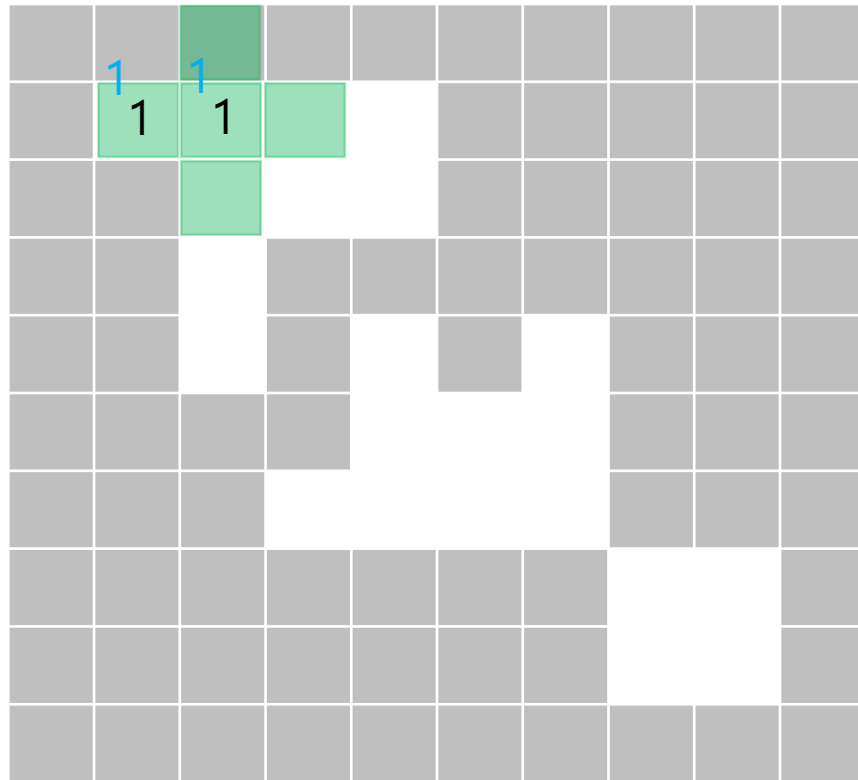
currentLabelCount = **1**

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



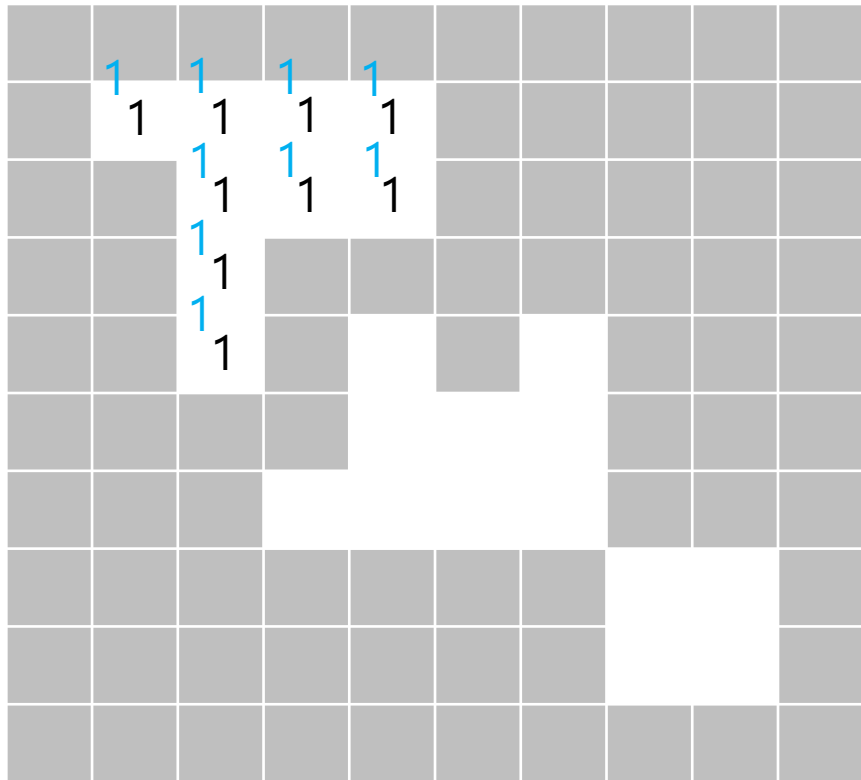
currentLabelCount = 1

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



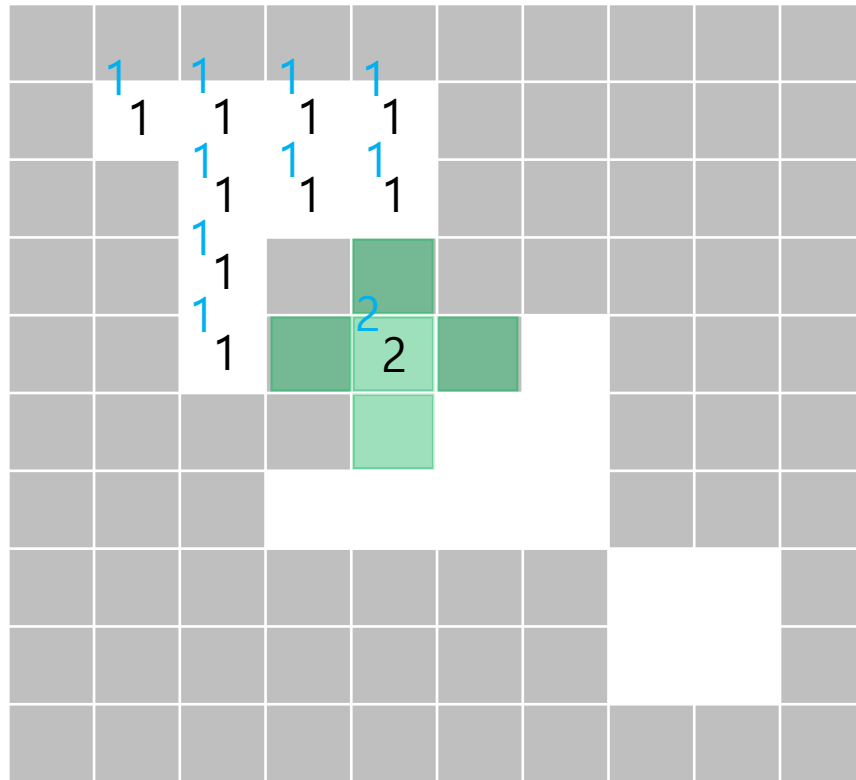
```
currentLabelCount = 1
```

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
}
// if no neighbor pixel assigned.
x.label = x.parent = (++currentLabelCount);
}
```


Algorithm for CCL (Connected-Component Labeling)



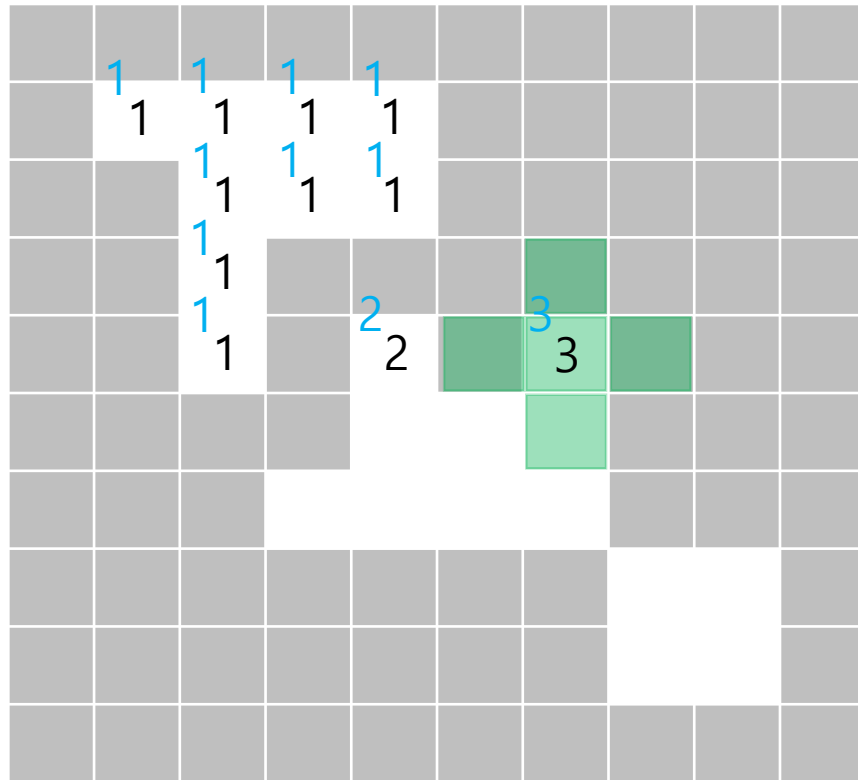
currentLabelCount = **2**

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)

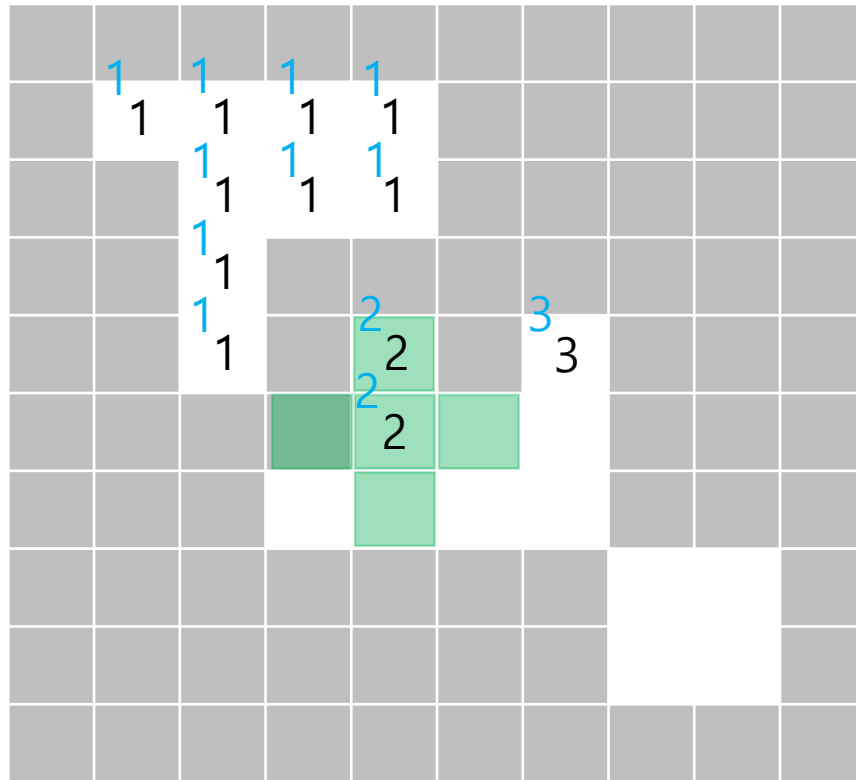


First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



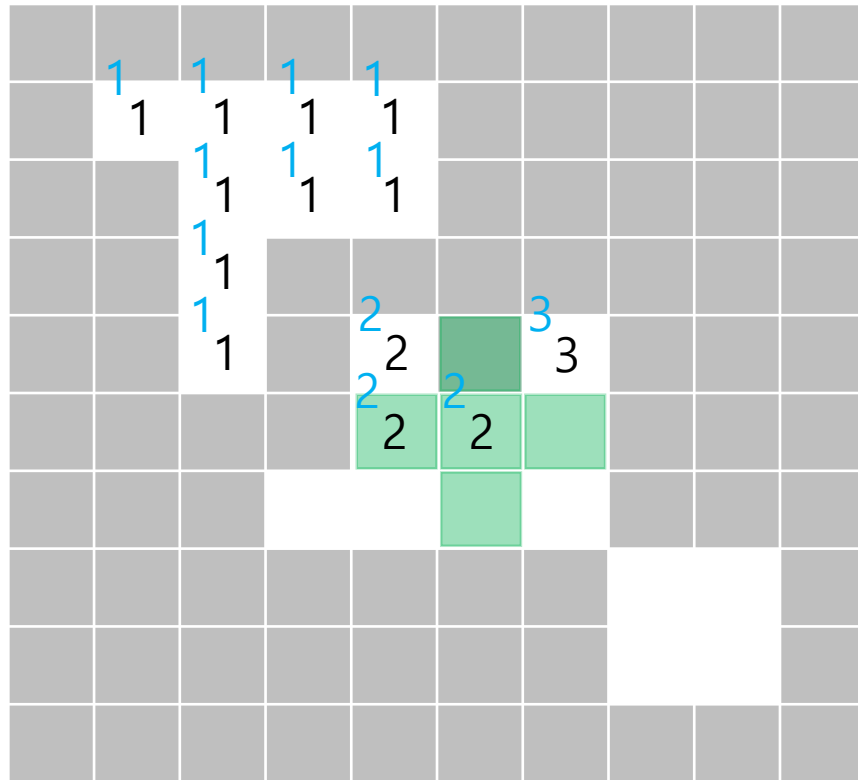
currentLabelCount = 3

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



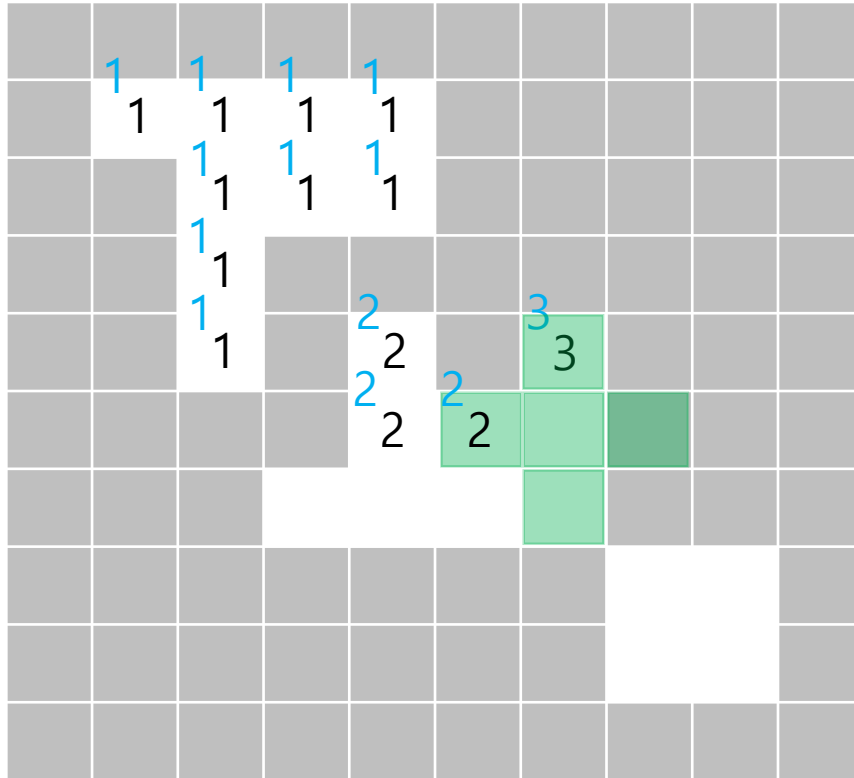
currentLabelCount = 3

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



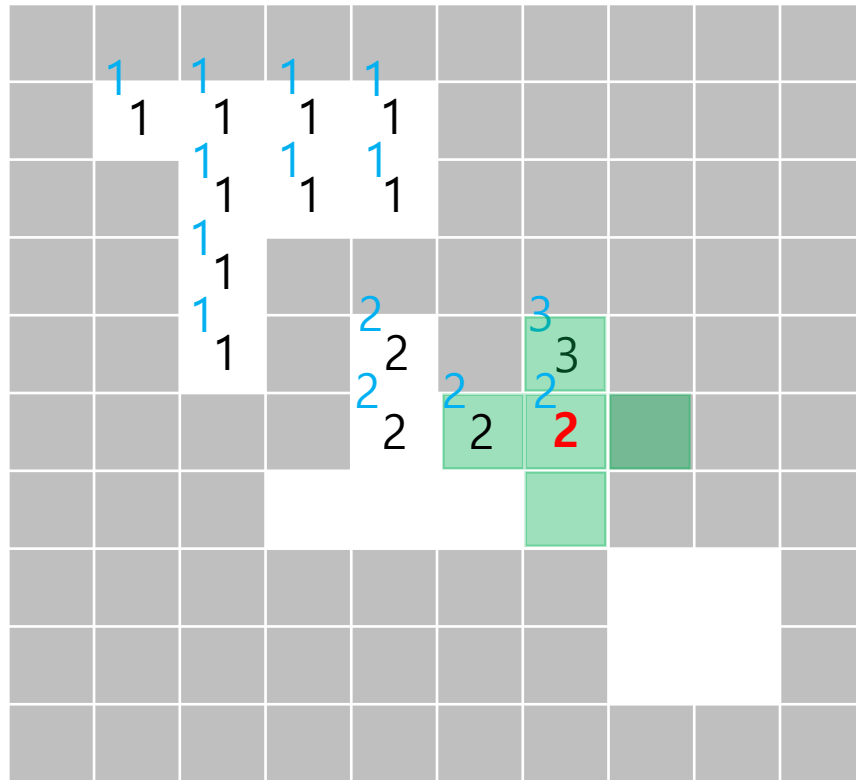
```
currentLabelCount = 3
```

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
}
// if no neighbor pixel assigned.
x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)

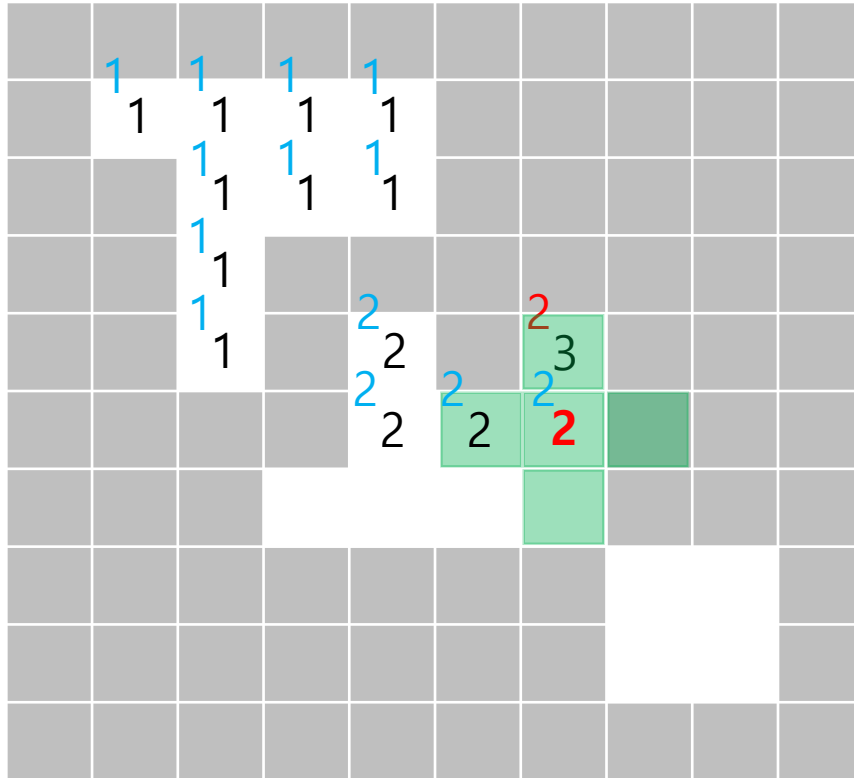


First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



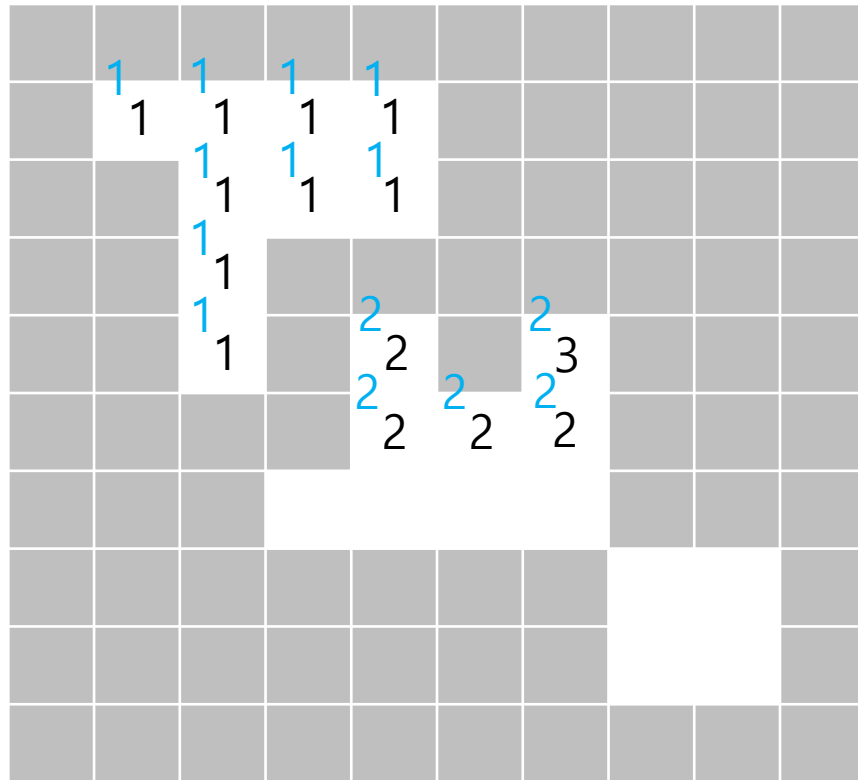
```
currentLabelCount = 3
```

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
}
// if no neighbor pixel assigned.
x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



currentLabelCount = 3

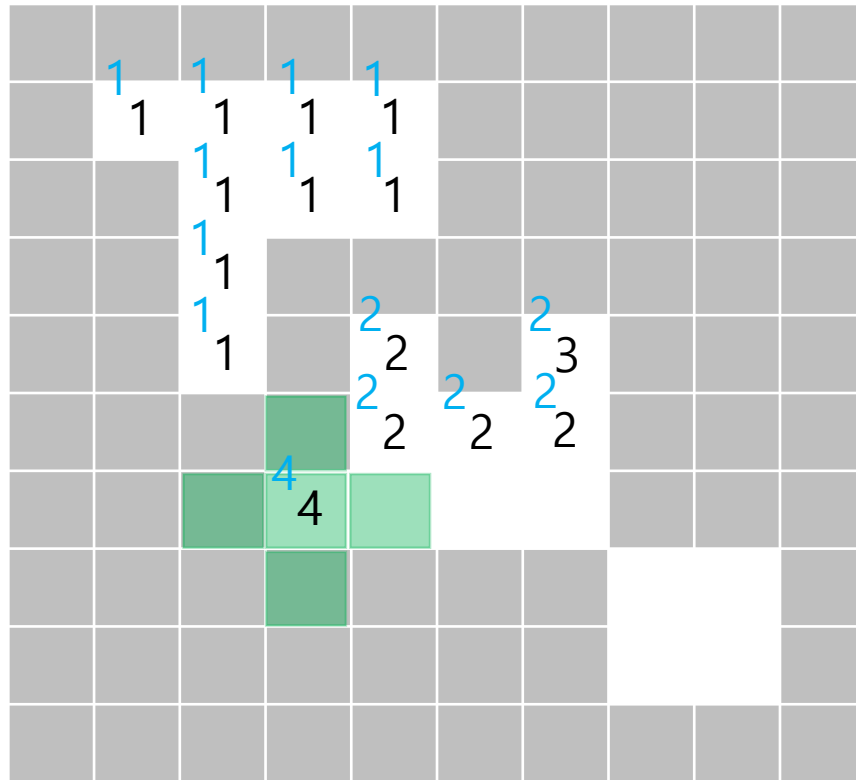
First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```


Algorithm for CCL (Connected-Component Labeling)

First pass (assigning labels)

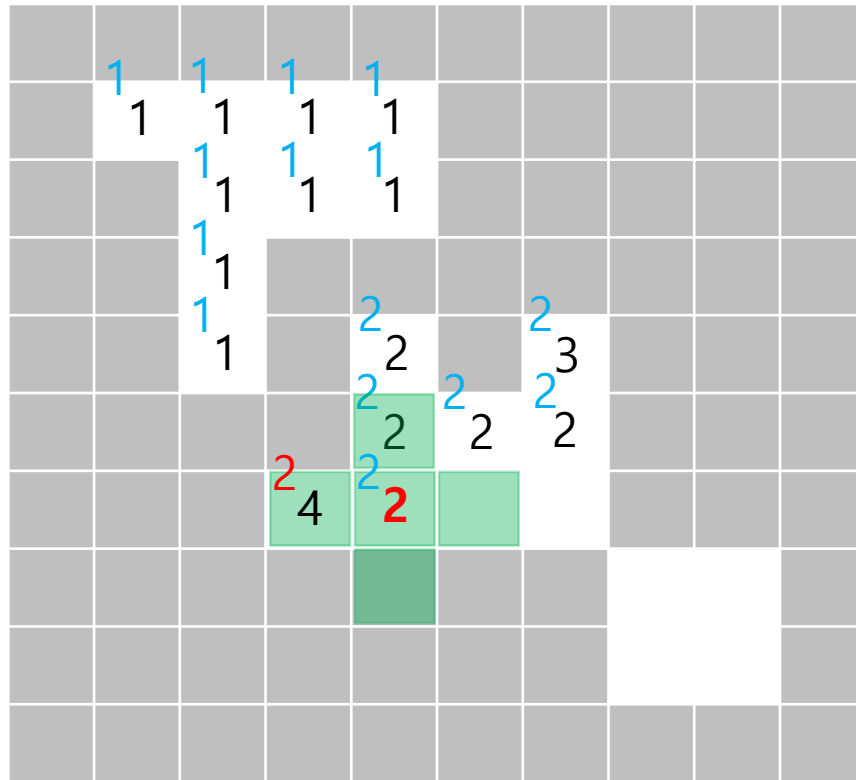


currentLabelCount = **4**

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



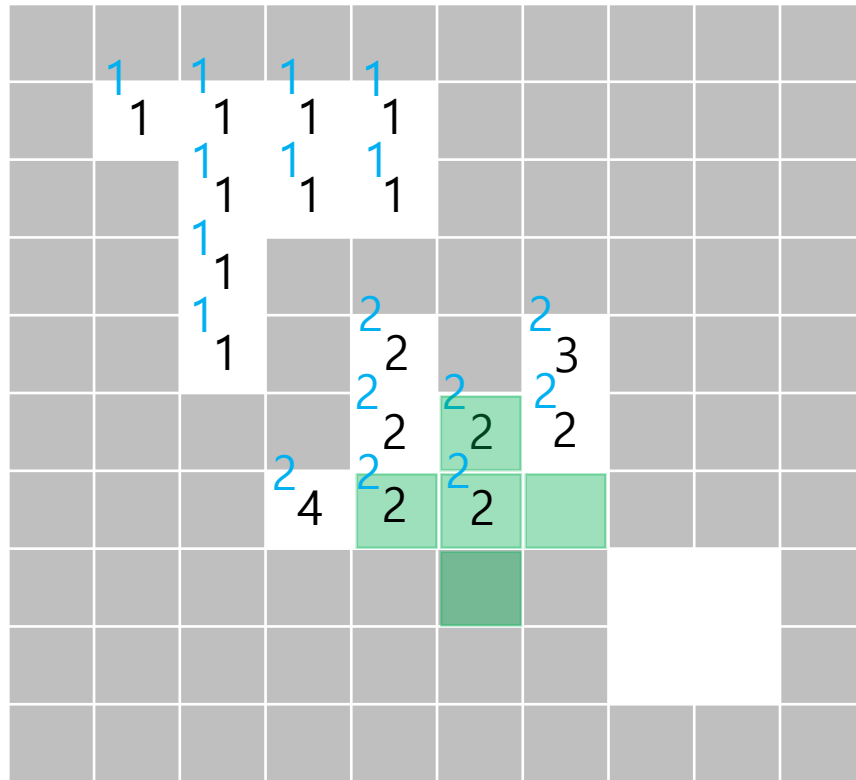
currentLabelCount = 4

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



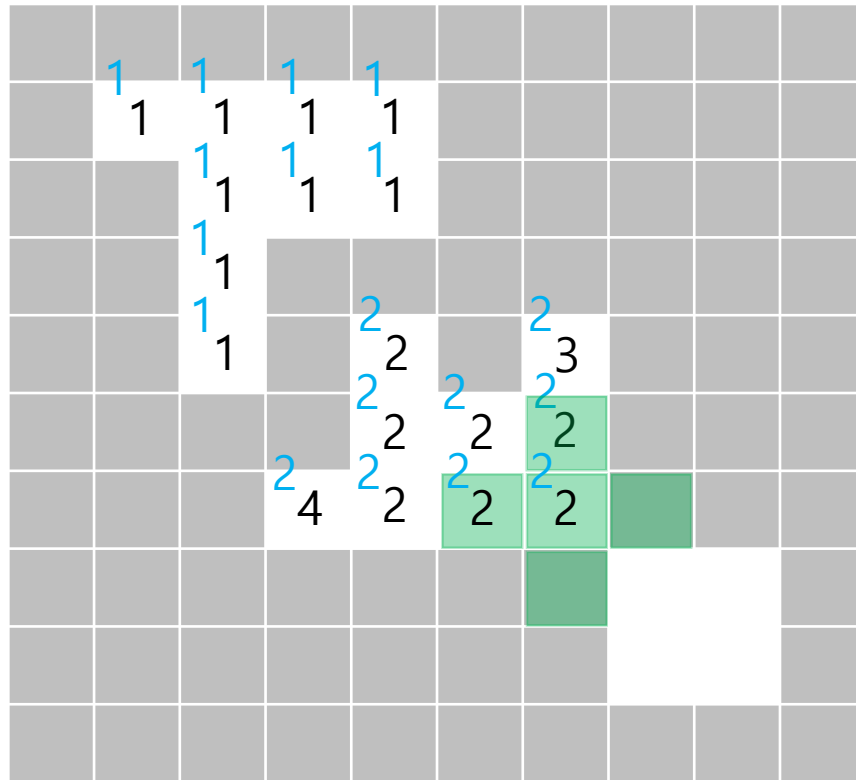
currentLabelCount = 4

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



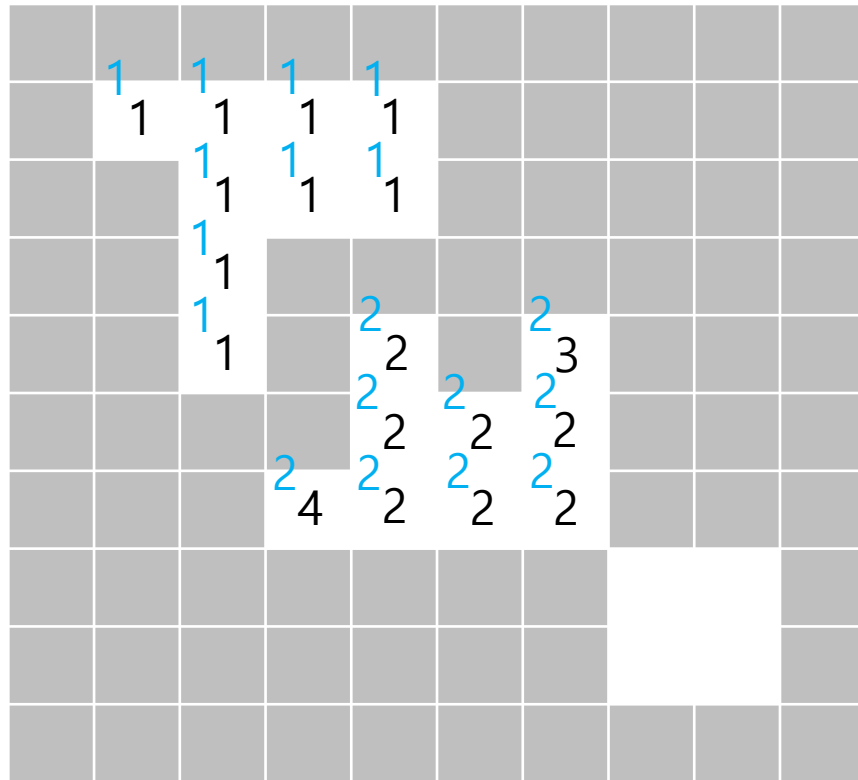
currentLabelCount = 4

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



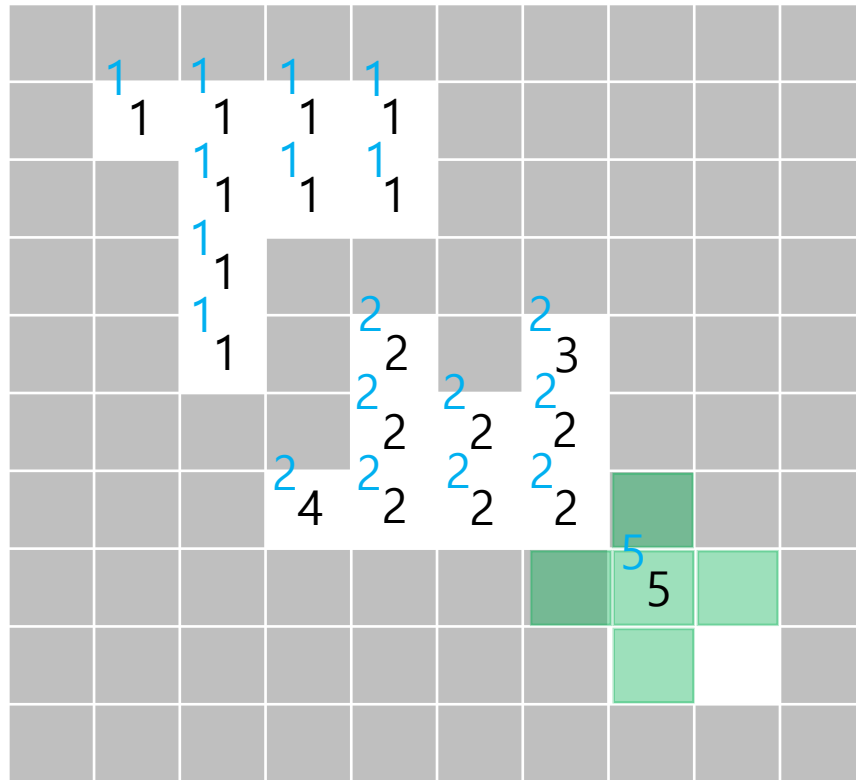
currentLabelCount = 4

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



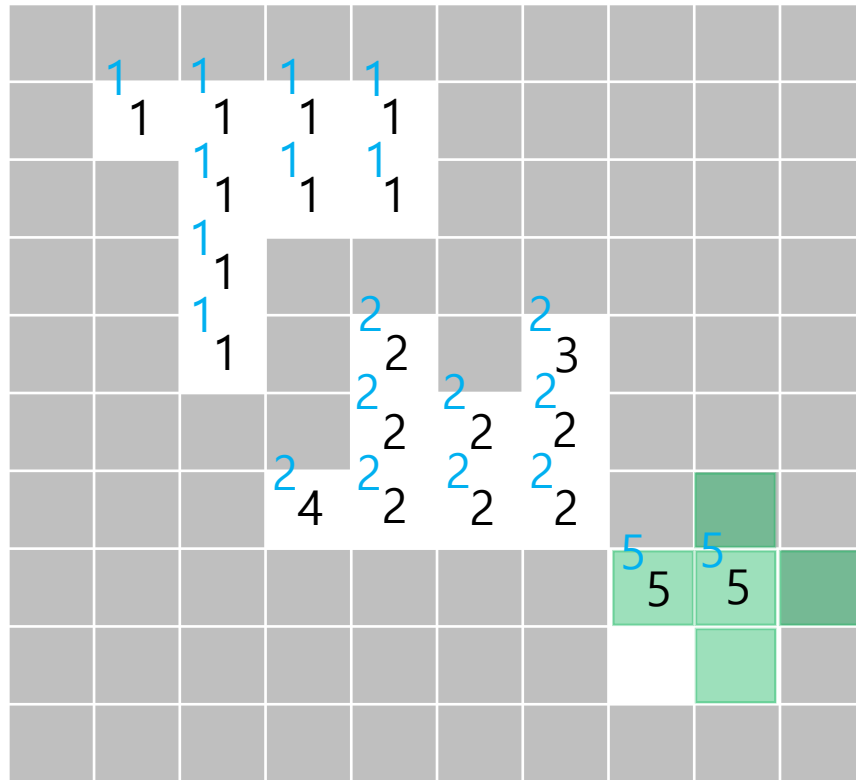
currentLabelCount = **5**

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



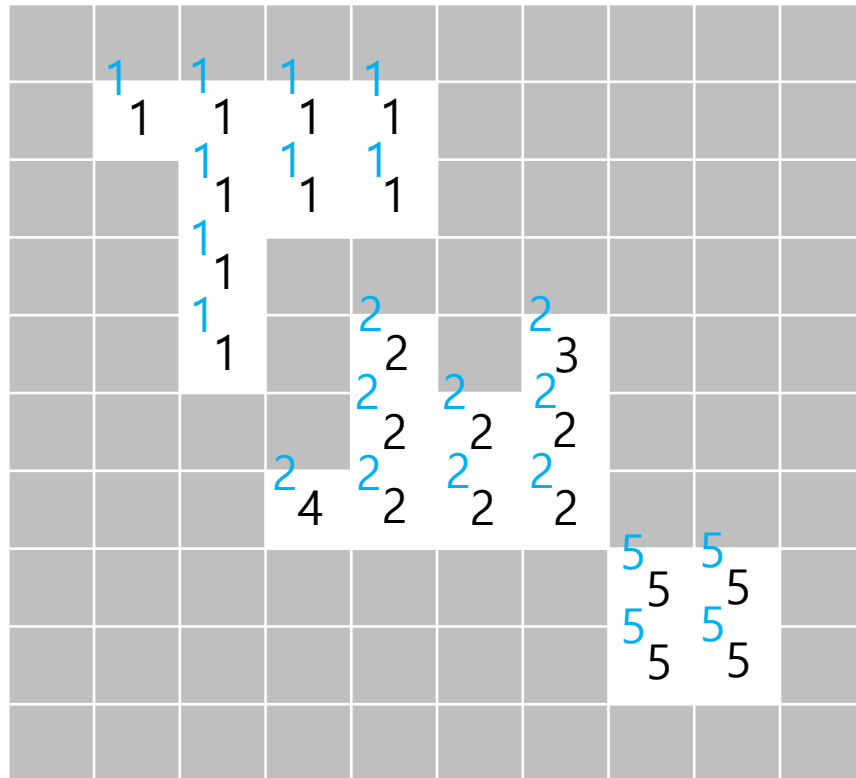
currentLabelCount = 5

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```

Algorithm for CCL (Connected-Component Labeling)



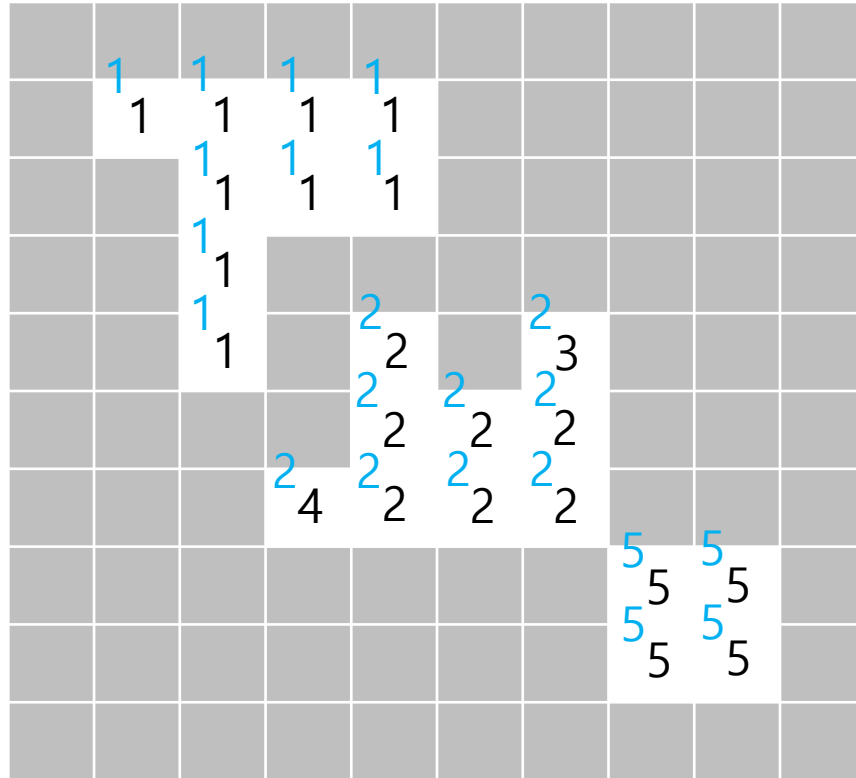
currentLabelCount = 5

First pass (assigning labels)

```
int currentLabelCount = 1;

for( x : every pixels ) {
    if( background ) continue;
    // process if foreground.
    for( y : every neighbor pixels ) {
        if( single label(s) obtained ) {
            x.label = y.label;
            x.parent = y.parent;
        }
        else { // neighbors have different labels
            Pixel z = find_min_label_pixel();
            x.label = z.label;
            x.parent = z.parent;
            Y.parent = z.parent;
        }
    }
    // if no neighbor pixel assigned.
    x.label = x.parent = (++currentLabelCount);
}
```


Algorithm for CCL (Connected-Component Labeling)

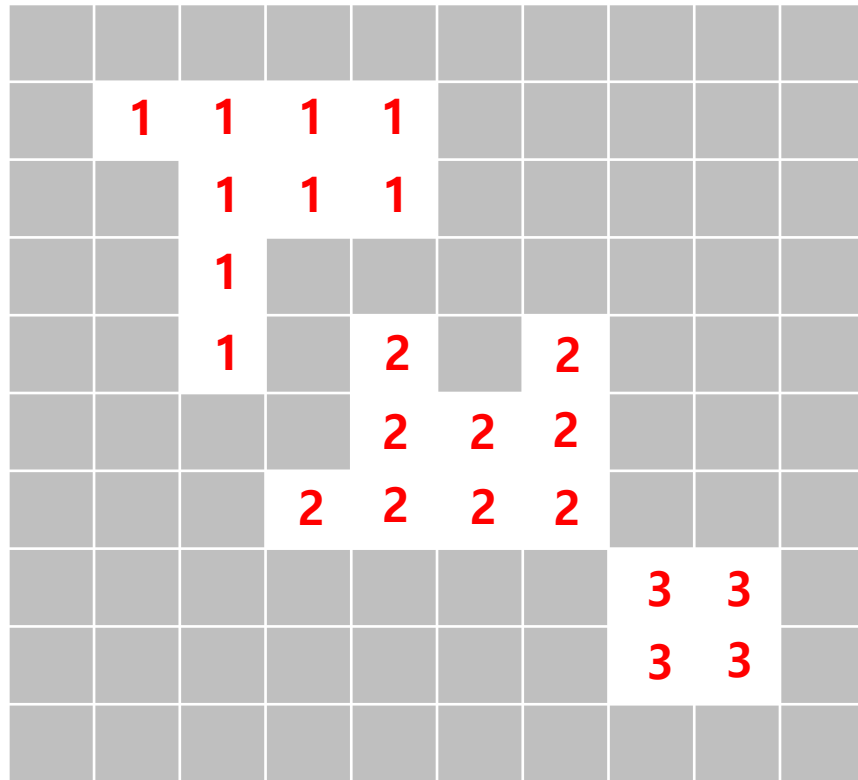


currentLabelCount = 5

Second pass (aggregation)

- Pass the image again pixel by pixel.
- Aggregate same parent pixels.
 - as same label.

Algorithm for CCL (Connected-Component Labeling)



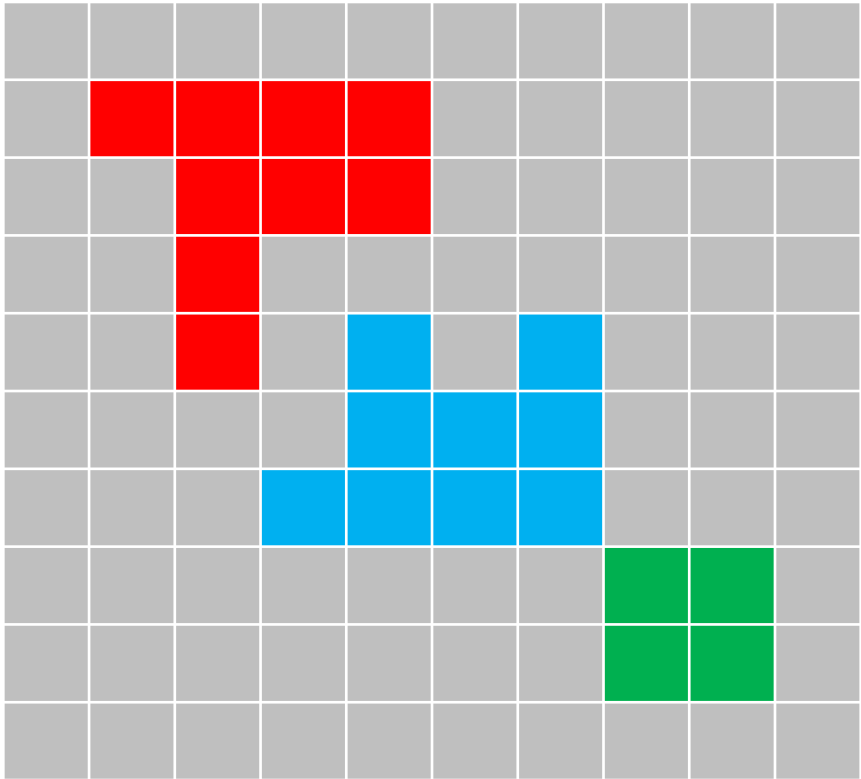
currentLabelCount = 5

1	→	1
2	→	2
5	→	3

Second pass (aggregation)

- Pass the image again pixel by pixel.
- Aggregate same parent pixels.
 - as same label.

Algorithm for CCL (Connected-Component Labeling)



Q & A