4.

The source codes for 4-neighbor and 8-neighbot are separated as question4.1.py and question4.2.

The basic thought was to scan every point in the matrix up to down and left to right, and set the points that *connects* with each other another in a certain “color”. In my code, I counted the cluster by the way, and the “color” was exactly the cluster number.

The details are included in the codes as annotations.

Input

10 (the number of the rows)

0 0 0 0 1 1 0 0 0 1 0 1 0 1 1 1 0 0 1 1

1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1 0 0

0 1 1 1 1 1 1 0 1 0 1 0 1 0 0 0 0 0 1 0

0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 1 0 1 1

1 0 0 1 0 0 0 0 0 0 1 1 1 0 1 1 1 0 1 1

1 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1

1 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 1

0 1 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 1

0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 0 1 1 1 0

1 0 1 0 1 0 1 1 1 1 0 0 1 0 1 0 0 0 0 1 (every row)

====================================================

Output

----------4 neighbors---------

0 0 0 0 6 6 0 0 0 10 0 12 0 15 15 15 0 0 20 20

1 0 4 0 0 0 0 0 10 10 0 0 0 0 15 0 0 19 0 0

0 4 4 4 4 4 4 0 10 0 11 0 14 0 0 0 0 0 17 0

0 0 0 0 4 4 0 0 0 0 11 0 0 0 17 17 17 0 17 17

2 0 0 2 0 0 0 0 0 0 11 11 11 0 17 17 17 0 17 17

2 2 2 2 0 7 0 0 0 0 0 0 0 16 0 0 17 17 0 17

2 2 0 0 0 0 8 0 0 0 0 13 13 0 0 0 0 17 0 17

0 2 0 0 0 8 8 0 0 0 0 13 13 0 0 0 0 17 17 17

0 0 5 5 5 0 0 9 0 9 0 0 13 0 0 0 17 17 17 0

3 0 5 0 5 0 9 9 9 9 0 0 13 0 18 0 0 0 0 21

----------8 neighbors---------

0 0 0 0 3 3 0 0 0 4 0 5 0 7 7 7 0 0 4 4

1 0 1 0 0 0 0 0 4 4 0 0 0 0 7 0 0 4 0 0

0 1 1 1 1 1 1 0 4 0 4 0 6 0 0 0 0 0 4 0

0 0 0 0 1 1 0 0 0 0 4 0 0 0 4 4 4 0 4 4

1 0 0 1 0 0 0 0 0 0 4 4 4 0 4 4 4 0 4 4

1 1 1 1 0 1 0 0 0 0 0 0 0 4 0 0 4 4 0 4

1 1 0 0 0 0 1 0 0 0 0 4 4 0 0 0 0 4 0 4

0 1 0 0 0 1 1 0 0 0 0 4 4 0 0 0 0 4 4 4

0 0 1 1 1 0 0 1 0 1 0 0 4 0 0 0 4 4 4 0

2 0 1 0 1 0 1 1 1 1 0 0 4 0 8 0 0 0 0 4