**Computer Vision HW7: Thinning**

**R10741015 鄭傑鴻**

**Oct. 28, 2022**

**Since some of the operations (i.e., thresholding, down-sampling and Yokoi) are identical to the code I designed in HW6, I’ll use them directly.**

**Step 1: Thresholding, Down Sampling and Generate Yokoi Connectivity Number**

* Description:

Call the functions implemented in HW6. Please kindly refer to the report of HW6 if you are interested.

* Result:

Down sampled Yokoi Connectivity Images (Value 0 replaced by ‘\_’ to better visualize)**:**

11111111\_\_\_\_\_\_\_\_12111111111122322221\_\_\_\_\_111111111111\_\_\_\_\_\_\_\_\_\_\_

15555551\_\_\_\_\_\_\_\_\_115555555511\_2\_11\_\_11\_\_\_1155555555511\_\_\_\_\_\_\_\_\_\_

15555551\_\_\_\_\_\_\_\_1\_2115555112\_\_21112221\_\_\_\_155555555551\_\_\_\_\_\_21\_\_

15555551\_\_\_\_\_\_\_\_1\_2\_155112\_22221511\_\_\_\_\_\_\_1555555555511\_\_\_\_\_1\_\_\_

15555551\_\_\_\_\_\_\_\_\_22\_2112\_22\_\_\_\_121\_\_\_\_\_\_\_\_15555555555511\_\_\_\_\_\_\_\_

15555551\_\_\_\_\_\_\_\_\_1\_\_2\_\_21\_2\_\_\_\_\_1\_\_\_1\_\_\_\_\_15555555555551\_\_\_\_\_\_\_\_

15555551\_\_\_\_\_\_\_\_\_\_\_12\_1\_\_121111\_\_\_\_1321\_\_\_155555555555511\_\_\_\_\_\_\_

15111551\_\_\_\_\_\_\_\_\_\_\_1322\_1155551111\_\_\_\_\_\_\_\_155555555555551\_\_\_\_\_\_\_

111\_1551\_\_\_\_\_\_\_\_\_\_\_\_1\_\_121555555511\_\_\_\_\_\_\_155555555555511\_\_\_\_\_\_\_

11\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_21155555511\_\_\_\_\_\_15511155555511\_\_\_\_\_\_\_\_

21\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2\_15555555111\_\_\_\_1551\_11555511\_\_\_\_\_\_\_\_\_

1\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2\_155555555511\_\_\_1551\_\_115551\_\_\_\_\_\_\_\_\_1

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1121155555555551\_\_\_1551\_\_\_15511\_\_\_\_\_\_\_\_12

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_15555555555555511\_\_1551\_\_\_1111\_\_\_\_\_\_\_\_111

\_\_\_\_1551\_\_\_\_\_\_\_\_1\_\_\_\_\_2221155555555555511\_1151\_\_\_\_11\_\_\_\_\_\_\_\_1151

\_\_\_\_1551\_\_\_\_\_\_\_\_2\_\_\_\_22\_1\_1555555555555511\_151\_\_11111\_\_\_\_\_\_\_1551

\_\_\_\_1551\_\_\_\_\_\_\_\_2\_\_\_\_1\_\_\_11555555555555551\_151\_115551\_\_\_\_\_\_11551

\_\_\_\_1551\_\_\_\_\_\_\_\_2\_\_\_\_\_\_\_11555555555555555111511155511\_\_\_\_\_115551

\_\_\_\_1551\_\_\_\_\_\_\_12\_\_\_\_\_\_11555555555555555555555555551\_\_\_\_\_\_155551

\_\_\_\_1551\_\_\_\_\_\_\_11\_\_\_\_\_221555555555555555555555555112\_\_\_\_\_1155551

\_\_\_\_1551\_\_\_\_\_\_\_111\_\_\_22\_15555555555555555555555551\_1\_\_\_\_\_1555551

\_\_\_\_1551\_\_\_\_\_\_\_1511\_\_1\_125112111112111555555555111\_\_\_\_\_\_11555551

\_\_\_\_1551\_\_\_\_\_\_\_15521\_\_1\_121\_1\_11\_\_1\_\_15555555111\_\_\_\_\_\_\_\_15555551

\_\_\_\_1551\_\_\_\_\_\_\_1151\_\_132\_2\_\_\_\_\_\_\_\_\_\_1155555111\_\_\_\_\_\_\_\_\_115555551

\_\_\_\_1551\_\_\_\_\_\_\_\_151\_\_\_\_322\_\_\_\_\_\_\_\_\_115555111\_\_121\_\_\_\_\_\_155555551

\_\_\_\_1551\_\_\_\_\_\_\_\_1221\_\_\_2\_\_\_\_\_\_\_\_\_\_\_1555551\_\_\_131\_\_\_\_\_\_1155555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_2\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_115555511\_\_\_1\_\_\_\_\_\_\_1155555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1155555551\_\_\_\_\_\_\_\_\_\_1\_155555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_2\_\_\_\_\_\_\_\_\_\_\_\_\_\_11555555551\_\_\_\_\_\_\_\_\_\_21155555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_115555555551\_\_\_\_\_\_\_\_\_\_15555555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_11511115555521\_\_1\_\_\_\_\_115555555551

\_\_\_\_1551\_\_\_\_\_\_\_\_1\_1\_\_\_\_\_\_\_\_\_\_11111\_\_1155511\_\_\_2\_\_\_\_\_155555555551

\_\_\_\_1551\_\_\_\_\_\_\_131\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_15111\_\_\_\_2\_\_\_\_\_155555555551

\_\_\_\_1551\_\_\_\_\_\_121\_\_\_\_\_\_\_\_\_\_1121\_\_\_1\_\_111\_\_1\_\_\_2\_\_\_\_1155555555551

\_\_\_\_1551\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_111\_1\_\_221\_11\_\_1\_\_\_2\_\_\_\_1555555555551

\_\_\_\_1551\_\_\_\_12\_\_\_\_\_\_\_1\_\_\_\_\_21\_121\_\_11\_1111\_\_\_\_2\_\_\_\_1555555555551

\_\_\_\_1551\_\_\_\_\_1\_\_\_\_\_\_12\_\_\_\_22\_\_151111111551\_\_\_\_2\_\_\_11555555555551

\_\_\_\_1551\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_2\_\_\_1555551115511\_\_\_1\_\_\_15555555555551

\_\_\_\_1551\_\_\_2\_\_\_\_\_\_\_\_\_\_\_\_\_22\_\_12555551\_15551\_\_\_\_1\_\_15555555555551

\_\_\_\_1551\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1555511\_11511\_\_\_\_2\_115555555555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_21\_\_\_\_\_155551\_1\_151\_\_\_\_\_2\_155555555555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2\_\_\_\_\_\_15555112\_151\_\_\_\_\_2\_155555555555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_1\_\_\_1\_1\_\_\_\_\_1155555511111\_\_\_\_\_2\_155555555555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_2\_\_22\_\_\_\_\_\_\_111511111212\_\_\_\_\_\_21155555555555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_1\_12\_\_\_\_\_\_\_\_\_\_151\_\_\_\_2\_1\_\_\_\_\_\_15555555111555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1111\_\_121\_\_\_\_\_\_\_155555551\_1555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111111\_\_\_\_\_\_\_155555551\_1555551

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_115551\_\_\_\_\_\_\_\_155555551\_1555511

\_\_\_\_1551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_15551\_\_\_\_\_\_\_\_211111111\_155511\_

\_\_\_\_11521\_\_\_\_\_\_1\_\_\_12\_\_\_\_\_\_\_\_\_\_122155511\_\_\_\_\_\_\_2\_\_\_\_\_11\_115511\_\_

1\_\_\_\_151\_\_\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_155555111\_\_\_\_\_2111\_\_\_\_\_15511\_\_\_

22\_\_\_1511\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_15555555111\_\_\_155111\_\_\_1511\_\_\_\_

\_22\_\_1511\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_15555555551\_\_\_155551\_\_1151\_\_\_\_\_

\_\_2\_\_151\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_11155555555511\_\_155511\_\_1511\_\_\_\_\_

\_\_2\_\_1521\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_155555555555511\_15551\_12151\_\_\_\_\_\_

\_\_2\_\_151\_\_\_\_\_\_\_\_\_\_\_121\_\_\_\_\_\_\_\_\_155555555555551\_155511\_1551\_\_\_\_\_\_

\_\_2\_\_1511\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_155555555555551\_115551\_1511\_\_\_\_\_\_

\_\_21\_1511\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_155555555555551\_\_111111151\_\_\_\_\_\_\_

\_\_11\_151\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11555555555555511\_\_\_\_111511\_\_\_\_\_\_\_

\_\_11\_151\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_15555555555555551\_\_\_\_\_\_151\_\_\_\_\_\_\_\_

\_\_11\_151\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_115555555555555551\_\_\_\_\_\_211\_\_\_\_\_\_\_\_

\_\_11\_151\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1155555555555555511\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_

\_\_11\_151\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_155555555555555551\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_11\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1211111111111111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 2: Implement the Pair Relationship Operator**

* Description

Starting from the Yokoi result in step 1, find the edge pixels (those with Yokoi connectivity number quals 1), check its 4-connected neighbors to decide whether they have edge neighbors. If yes, label it as ‘p’: interesting. For the other scenarios, assign ‘q’: not interesting.

* Algorithm

*Scan through all the pixels:*

*If the pixel is an edge:*

*If it has edge neighbors (function h, f): assign ‘p’*

*Else: assign ‘q’*

*If not an edge:*

*Assign ‘q’*

* Code

def pairRelationship(self, array):  
 # Input: Yokoi (shape 64,64, integer)  
 # Output: Pair Relationship result (shape 64,64, integer)  
 def funH(a, m):  
 if a == m:  
 return 1  
 else:  
 return 0  
  
 def funF(a1, a2, a3, a4):  
 if (a1 + a2 + a3 + a4) < 1:  
 return 'q'  
 else:  
 return 'p'  
  
 ret\_array = np.zeros((64,64)).astype(str)  
 for i in range(64):  
 for j in range(64):  
 if array[i][j] == 1:  
 x0 = array[i][j]  
 x1 = 0 if not self.inRange(i + 0, j + 1) else array[i + 0][j + 1]  
 x2 = 0 if not self.inRange(i - 1, j + 0) else array[i - 1][j + 0]  
 x3 = 0 if not self.inRange(i + 0, j - 1) else array[i + 0][j - 1]  
 x4 = 0 if not self.inRange(i + 1, j + 0) else array[i + 1][j + 0]  
  
 a1 = funH(x0, x1)  
 a2 = funH(x0, x2)  
 a3 = funH(x0, x3)  
 a4 = funH(x0, x4)  
  
 ret\_array[i][j] = funF(a1, a2, a3, a4)  
 elif array[i][j] == 0:  
 ret\_array[i][j] = '0'  
 else:  
 ret\_array[i][j] = 'q'  
 return ret\_array

* Result

Take the result from the first iteration as an example:

pppppppp\_\_\_\_\_\_\_\_qqppppppppppqqqqqqqq\_\_\_\_\_pppppppppppp\_\_\_\_\_\_\_\_\_\_\_

pqqqqqqp\_\_\_\_\_\_\_\_\_ppqqqqqqqqpp\_q\_pp\_\_pp\_\_\_ppqqqqqqqqqpp\_\_\_\_\_\_\_\_\_\_

pqqqqqqp\_\_\_\_\_\_\_\_p\_qppqqqqppq\_\_qpppqqqp\_\_\_\_pqqqqqqqqqqp\_\_\_\_\_\_qq\_\_

pqqqqqqp\_\_\_\_\_\_\_\_p\_q\_pqqppq\_qqqqpqpp\_\_\_\_\_\_\_pqqqqqqqqqqpp\_\_\_\_\_q\_\_\_

pqqqqqqp\_\_\_\_\_\_\_\_\_qq\_qppq\_qq\_\_\_\_pqp\_\_\_\_\_\_\_\_pqqqqqqqqqqqpp\_\_\_\_\_\_\_\_

pqqqqqqp\_\_\_\_\_\_\_\_\_q\_\_q\_\_qq\_q\_\_\_\_\_q\_\_\_q\_\_\_\_\_pqqqqqqqqqqqqp\_\_\_\_\_\_\_\_

pqqqqqqp\_\_\_\_\_\_\_\_\_\_\_pq\_q\_\_pqpppp\_\_\_\_qqqq\_\_\_pqqqqqqqqqqqqpp\_\_\_\_\_\_\_

pqpppqqp\_\_\_\_\_\_\_\_\_\_\_pqqq\_ppqqqqpppp\_\_\_\_\_\_\_\_pqqqqqqqqqqqqqp\_\_\_\_\_\_\_

ppp\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_q\_\_qqpqqqqqqqpp\_\_\_\_\_\_\_pqqqqqqqqqqqqpp\_\_\_\_\_\_\_

pp\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_qppqqqqqqpp\_\_\_\_\_\_pqqpppqqqqqqpp\_\_\_\_\_\_\_\_

qp\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_q\_pqqqqqqqppp\_\_\_\_pqqp\_ppqqqqpp\_\_\_\_\_\_\_\_\_

q\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_q\_pqqqqqqqqqpp\_\_\_pqqp\_\_ppqqqp\_\_\_\_\_\_\_\_\_q

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ppqppqqqqqqqqqqp\_\_\_pqqp\_\_\_pqqpp\_\_\_\_\_\_\_\_pq

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pqqqqqqqqqqqqqqpp\_\_pqqp\_\_\_pppp\_\_\_\_\_\_\_\_ppp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_q\_\_\_\_\_qqqppqqqqqqqqqqqqpp\_ppqp\_\_\_\_pp\_\_\_\_\_\_\_\_ppqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_q\_\_\_\_qq\_q\_pqqqqqqqqqqqqqpp\_pqp\_\_ppppp\_\_\_\_\_\_\_pqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_q\_\_\_\_q\_\_\_ppqqqqqqqqqqqqqqp\_pqp\_ppqqqp\_\_\_\_\_\_ppqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_q\_\_\_\_\_\_\_ppqqqqqqqqqqqqqqqpppqpppqqqpp\_\_\_\_\_ppqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_pq\_\_\_\_\_\_ppqqqqqqqqqqqqqqqqqqqqqqqqqqp\_\_\_\_\_\_pqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_pp\_\_\_\_\_qqpqqqqqqqqqqqqqqqqqqqqqqqqppq\_\_\_\_\_ppqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_ppp\_\_\_qq\_pqqqqqqqqqqqqqqqqqqqqqqqqp\_q\_\_\_\_\_pqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_pqpp\_\_q\_qqqppqpppppqpppqqqqqqqqqppp\_\_\_\_\_\_ppqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_pqqqq\_\_q\_qqp\_q\_pp\_\_q\_\_pqqqqqqqppp\_\_\_\_\_\_\_\_pqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_ppqp\_\_qqq\_q\_\_\_\_\_\_\_\_\_\_ppqqqqqppp\_\_\_\_\_\_\_\_\_ppqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_pqp\_\_\_\_qqq\_\_\_\_\_\_\_\_\_ppqqqqppp\_\_qqq\_\_\_\_\_\_pqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_pqqq\_\_\_q\_\_\_\_\_\_\_\_\_\_\_pqqqqqp\_\_\_qqq\_\_\_\_\_\_ppqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_q\_\_\_\_\_q\_\_\_\_\_\_\_\_\_\_ppqqqqqpp\_\_\_q\_\_\_\_\_\_\_ppqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_q\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ppqqqqqqqp\_\_\_\_\_\_\_\_\_\_q\_pqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_q\_\_\_\_\_\_\_\_\_\_\_\_\_\_ppqqqqqqqqp\_\_\_\_\_\_\_\_\_\_qppqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_q\_\_\_\_\_\_\_\_\_\_\_\_\_ppqqqqqqqqqp\_\_\_\_\_\_\_\_\_\_pqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_p\_\_\_\_\_\_\_\_\_\_\_ppqppppqqqqqqq\_\_q\_\_\_\_\_ppqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_q\_p\_\_\_\_\_\_\_\_\_\_ppppp\_\_ppqqqpp\_\_\_q\_\_\_\_\_pqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_qqq\_\_\_\_\_\_\_\_\_\_\_ppp\_\_\_\_\_pqppp\_\_\_\_q\_\_\_\_\_pqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_pqq\_\_\_\_\_\_\_\_\_\_ppqp\_\_\_q\_\_ppp\_\_p\_\_\_q\_\_\_\_ppqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_pp\_\_\_\_\_\_\_\_\_\_\_ppp\_q\_\_qqp\_pp\_\_p\_\_\_q\_\_\_\_pqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_qq\_\_\_\_\_\_\_q\_\_\_\_\_qp\_pqp\_\_pp\_pppp\_\_\_\_q\_\_\_\_pqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_q\_\_\_\_\_\_qq\_\_\_\_qq\_\_pqpppppppqqp\_\_\_\_q\_\_\_ppqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_q\_\_\_\_\_\_\_\_\_\_\_\_\_\_q\_\_\_pqqqqqpppqqpp\_\_\_q\_\_\_pqqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_q\_\_\_\_\_\_\_\_\_\_\_\_\_qq\_\_qqqqqqqp\_pqqqp\_\_\_\_q\_\_pqqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_q\_\_\_\_\_\_\_\_\_\_\_\_\_q\_\_\_\_pqqqqpp\_ppqpp\_\_\_\_q\_ppqqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_qq\_\_\_\_\_pqqqqp\_q\_pqp\_\_\_\_\_q\_pqqqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_q\_\_\_\_\_\_pqqqqppq\_pqp\_\_\_\_\_q\_pqqqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_q\_\_\_q\_q\_\_\_\_\_ppqqqqqqppppp\_\_\_\_\_q\_pqqqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_q\_\_qq\_\_\_\_\_\_\_pppqpppppqpq\_\_\_\_\_\_qppqqqqqqqqqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_q\_qq\_\_\_\_\_\_\_\_\_\_pqp\_\_\_\_q\_q\_\_\_\_\_\_pqqqqqqqpppqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pppp\_\_pqp\_\_\_\_\_\_\_pqqqqqqqp\_pqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pppppppp\_\_\_\_\_\_\_pqqqqqqqp\_pqqqqqp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ppqqqp\_\_\_\_\_\_\_\_pqqqqqqqp\_pqqqqpp

\_\_\_\_pqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pqqqp\_\_\_\_\_\_\_\_qpppppppp\_pqqqpp\_

\_\_\_\_ppqqq\_\_\_\_\_\_p\_\_\_qq\_\_\_\_\_\_\_\_\_\_qqqpqqqpp\_\_\_\_\_\_\_q\_\_\_\_\_pp\_ppqqpp\_\_

q\_\_\_\_pqp\_\_\_\_\_\_\_p\_\_\_\_q\_\_\_\_\_\_\_\_\_\_\_\_pqqqqqppp\_\_\_\_\_qppp\_\_\_\_\_pqqpp\_\_\_

qq\_\_\_pqpp\_\_\_\_\_\_\_\_\_\_p\_\_\_\_\_\_\_\_\_\_\_\_\_pqqqqqqqppp\_\_\_pqqppp\_\_\_pqpp\_\_\_\_

\_qq\_\_pqpp\_\_\_\_\_\_\_\_\_\_p\_\_\_\_\_\_\_\_\_\_\_\_\_pqqqqqqqqqp\_\_\_pqqqqp\_\_ppqp\_\_\_\_\_

\_\_q\_\_pqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_p\_\_\_\_\_\_\_\_pppqqqqqqqqqpp\_\_pqqqpp\_\_pqpp\_\_\_\_\_

\_\_q\_\_pqqq\_\_\_\_\_\_\_\_\_\_\_\_\_p\_\_\_\_\_\_\_\_pqqqqqqqqqqqqpp\_pqqqp\_qqpqp\_\_\_\_\_\_

\_\_q\_\_pqp\_\_\_\_\_\_\_\_\_\_\_qqq\_\_\_\_\_\_\_\_\_pqqqqqqqqqqqqqp\_pqqqpp\_pqqp\_\_\_\_\_\_

\_\_q\_\_pqpp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pqqqqqqqqqqqqqp\_ppqqqp\_pqpp\_\_\_\_\_\_

\_\_qp\_pqpp\_\_\_\_\_\_\_\_\_\_\_\_pp\_\_\_\_\_\_\_\_pqqqqqqqqqqqqqp\_\_pppppppqp\_\_\_\_\_\_\_

\_\_pp\_pqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ppqqqqqqqqqqqqqpp\_\_\_\_pppqpp\_\_\_\_\_\_\_

\_\_pp\_pqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pqqqqqqqqqqqqqqqp\_\_\_\_\_\_pqp\_\_\_\_\_\_\_\_

\_\_pp\_pqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ppqqqqqqqqqqqqqqqp\_\_\_\_\_\_qpp\_\_\_\_\_\_\_\_

\_\_pp\_pqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ppqqqqqqqqqqqqqqqpp\_\_\_\_\_q\_\_\_\_\_\_\_\_\_\_

\_\_pp\_pqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pqqqqqqqqqqqqqqqqp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_pp\_ppp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_qqppppppppppppppppp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 3: Implement the Connected Shrink Operator**

* Description:

By using the original binary image and the result from Pair Relationship operator, for those pixels who are marked as interesting (‘p’) by the pair relationship operator, check its neighbors in the original binary image to decide whether it can be eliminated without disconnecting the region.

Since Connected Shrink is a recursive operator, we must update the result in real time.

* Algorithm:

*Scan through all the pixels:*

*If the pixel is marked as interesting (‘p’):*

*If it can be eliminated (by using function f and h):*

*Update the pixel to background*

*Else: Remain unchanged*

*Else: Remain unchanged*

* Code:

def connectedShrink(self, pr\_array, binds\_array):  
 # Input: Pair Relationship(shape (64,64), str): '0':background, 'p': interesting, 'q':not interestin  
 # Binary Down-sample array(shape (64,64), int): 0:background, 1:foreground  
 # Output: Connected Shrink result(shape (64,64), int): 0:background, 1:foreground  
 def funH(b,c,d,e):  
 if b == c and (d!=b or e!=b): return 1  
 else: return 0  
 def funF(a1,a2,a3,a4):  
 if (a1+a2+a3+a4) == 1: # make it background  
 self.cnt += 1  
 return 0  
 else:  
 return 1  
 ret\_array = np.copy(binds\_array)  
 for i in range(64):  
 for j in range(64):  
 if pr\_array[i][j] == 'p':  
 x0 = ret\_array[i][j]  
 x1 = 0 if not self.inRange(i + 0, j + 1) else ret\_array[i + 0][j + 1]  
 x2 = 0 if not self.inRange(i - 1, j + 0) else ret\_array[i - 1][j + 0]  
 x3 = 0 if not self.inRange(i + 0, j - 1) else ret\_array[i + 0][j - 1]  
 x4 = 0 if not self.inRange(i + 1, j + 0) else ret\_array[i + 1][j + 0]  
 x5 = 0 if not self.inRange(i + 1, j + 1) else ret\_array[i + 1][j + 1]  
 x6 = 0 if not self.inRange(i - 1, j + 1) else ret\_array[i - 1][j + 1]  
 x7 = 0 if not self.inRange(i - 1, j - 1) else ret\_array[i - 1][j - 1]  
 x8 = 0 if not self.inRange(i + 1, j - 1) else ret\_array[i + 1][j - 1]  
  
 # Upper right: [0,0], [0, 1], [-1, 1], [-1, 0]  
 a1 = funH(x0, x1, x6, x2)  
 # Upper left: [0,0], [-1, 0], [-1, -1], [0, -1]  
 a2 = funH(x0, x2, x7, x3)  
 # Bottom left: [0,0], [0, -1], [1, -1], [1, 0]  
 a3 = funH(x0, x3, x8, x4)  
 # Bottom right: [0,0], [1, 0], [1, 1], [0, 1]  
 a4 = funH(x0, x4, x5, x1)  
 ret\_array[i][j] = funF(a1, a2, a3, a4)  
 return ret\_array

* Result:

Take the result from the first iteration as an example:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_11111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_1

\_111111\_\_\_\_\_\_\_\_\_\_111111111111\_1\_\_1\_\_\_\_\_\_\_\_\_111111111\_\_\_\_\_\_\_\_\_\_1\_

\_111111\_\_\_\_\_\_\_\_\_\_\_1\_\_1111\_\_1\_\_1\_\_1111\_\_\_\_\_\_1111111111\_\_\_\_\_\_\_11\_\_

\_111111\_\_\_\_\_\_\_\_\_1\_1\_\_11\_11\_1111111\_\_\_\_\_\_\_\_\_1111111111\_\_\_\_\_\_\_1\_\_\_

\_111111\_\_\_\_\_\_\_\_\_\_11\_1111\_11\_\_\_\_\_1\_\_1\_1\_\_\_\_\_11111111111\_\_\_\_\_1\_\_\_\_

\_111111\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_11\_1\_\_\_\_\_1\_\_\_1\_\_\_\_\_\_111111111111\_\_\_1\_\_\_\_\_

\_111111\_\_\_\_\_\_\_\_\_\_\_\_\_1\_1\_\_\_1\_\_\_\_\_\_\_\_1111\_\_\_\_111111111111\_\_\_\_\_\_\_\_\_

\_1\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_1111111111111\_\_\_\_\_\_\_\_

\_1\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_1111111111\_\_\_\_\_\_\_\_\_\_1111111111111\_\_\_\_\_\_\_\_

\_1\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_111111\_\_\_\_\_\_\_\_\_11\_\_\_111111\_\_\_\_\_\_\_\_\_\_

11\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_1111111\_\_\_\_\_\_\_\_11\_\_\_\_11111\_\_\_\_\_\_\_\_\_\_

1\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_111111111\_\_\_\_\_\_11\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_1

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_1111111111\_\_\_\_\_11\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_1

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111111111111\_\_\_\_\_11\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_\_\_111\_\_111111111111\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_11

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_\_11\_1\_\_1111111111111\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_11\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_11111111111111\_\_\_1\_\_\_\_111\_\_\_\_\_\_\_\_\_11\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_111111111111111\_\_\_1\_\_\_1111\_\_\_\_\_\_\_\_111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_11111111111111111111111111\_\_\_\_\_\_\_\_1111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_1\_111111111111111111111111111\_11\_\_\_\_\_\_\_1111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_\_11\_\_111111111111111111111111\_\_1\_\_\_\_\_\_11111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_\_1\_111\_\_1\_\_1\_\_1\_\_\_111111111\_1\_\_\_\_\_\_\_\_\_11111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1111\_\_1\_11\_\_1\_\_\_\_\_1\_\_\_1111111\_1\_\_\_1\_\_\_\_\_\_111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_\_111\_1\_\_\_\_\_\_\_\_\_\_\_\_11111\_1\_\_\_\_1\_\_\_\_\_\_\_111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_111\_\_\_\_\_\_\_\_\_\_\_1111\_1\_\_\_111\_\_\_\_\_\_\_1111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_111\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_11111\_\_\_\_111\_\_\_\_\_\_\_\_1111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_1\_\_\_\_\_\_\_\_\_\_\_\_11111\_\_\_\_\_1\_\_\_\_\_\_\_\_\_1111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_1111111\_\_\_1\_\_\_\_\_\_\_1\_\_1111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111111\_\_\_\_\_\_\_\_\_\_\_1\_\_1111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_\_\_\_\_\_\_\_\_\_\_111111111\_\_\_\_\_\_\_\_\_\_\_1111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_1111111\_\_1\_\_\_\_\_\_\_111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_1\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_111\_\_\_\_\_1\_\_\_\_\_\_1111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_1\_1\_\_\_\_\_1\_\_\_\_\_\_1111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_11\_1\_\_\_\_\_\_\_\_\_\_11\_\_\_1\_\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_\_1111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_1\_\_11\_\_1\_\_\_1\_\_\_1\_\_\_\_\_11111111111\_

\_\_\_\_\_11\_\_\_\_\_11\_\_1\_\_\_\_1\_\_\_\_\_11\_\_1\_\_\_1\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_11111111111\_

\_\_\_\_\_11\_\_\_\_\_\_1\_\_\_\_\_\_11\_\_\_\_11\_\_\_1\_\_\_1\_\_111\_\_\_\_\_1\_\_\_\_\_11111111111\_

\_\_\_\_\_11\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1111111111\_\_\_\_\_1\_\_\_\_111111111111\_

\_\_\_\_\_11\_\_\_\_1\_\_\_\_1\_\_\_\_\_1\_\_11\_\_1111111\_\_\_111\_\_\_\_\_1\_\_\_111111111111\_

\_\_\_\_\_11\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_11111\_\_\_\_11\_\_\_\_\_1\_\_\_111111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_1\_1\_\_\_\_\_\_11\_\_\_\_\_\_1111\_\_1\_\_1\_\_\_\_\_\_1\_\_1111111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_1111\_\_1\_\_1\_\_\_\_\_\_1\_\_1111111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_\_1\_1\_\_\_\_\_\_\_1111111\_\_1\_\_\_\_\_\_1\_\_1111111111111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1111\_\_\_\_\_\_1\_\_1111111111111\_

\_\_\_\_\_11\_\_\_1\_\_\_\_\_\_1\_11\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_1\_1\_\_\_\_\_\_11111111\_\_\_11111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_1\_\_1\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_1\_\_\_\_\_\_\_\_\_1111111\_\_\_11111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_1\_\_\_\_\_\_\_\_\_1111111\_\_\_11111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111111\_\_\_\_\_\_\_\_\_1111111\_\_\_11111\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_11\_\_\_\_\_1\_\_\_111\_\_\_

\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_111\_111\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_111\_\_\_

1\_\_\_\_\_1\_\_1\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_111111\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_11\_\_\_\_\_

11\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1111111\_\_\_\_\_\_111\_\_\_\_\_\_\_11\_\_\_\_\_

\_11\_\_\_11\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_111111111\_\_\_\_\_1111\_\_\_\_\_1\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_111111111\_\_\_\_\_1111\_\_\_\_11\_\_\_\_\_\_

\_\_1\_\_\_111\_\_\_\_1\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_111111111111\_\_\_\_111\_\_11\_1\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_1111111111111\_\_\_111\_\_\_111\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_1111111111111\_\_\_\_111\_\_\_11\_\_\_\_\_\_\_

\_\_1\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_1111111111111\_\_\_\_\_\_1\_\_\_1\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1111111111111\_\_\_\_\_\_11111\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111111111111111\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_111111111111111\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111111111111111\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_1111111111111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 4: Repeat the above procedure until no change**

* Description:

Repeat the Yokoi->Pair Relationship->Connected Shrink procedure, until it can’t be further skeletonized (7 times later).

* Algorithm:

*Set up a variable ‘cnt’ to keep track of whether there are changes*

*While cnt != 0:*

*Repeat the Yokoi->Pair Relationship->Connected Shrink procedure*

*Show the final result*

* Code:

def thinning(self):  
  
 ans = np.copy(self.binds)  
 while(self.cnt != 0):  
 print('Iteration {}'.format(self.iter))  
 # Initialize  
 self.cnt = 0  
  
 # Execute  
 yk = self.yokoi(ans)  
 self.printImg(yk)  
 pr = self.pairRelationship(yk)  
 self.printImg(pr)  
 sh = self.connectedShrink(pr, ans)  
 self.printImg(sh)  
  
 # Update  
 ans = sh  
 self.iter += 1  
 self.showImg(ans, save=True)

* Result:

After looping for 7 times:

 (The result with shape (64, 64) in .bmp)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_11111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111\_\_11111\_1\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_1111\_\_1\_\_1\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_1\_\_\_1\_11\_1111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_1111\_11\_\_\_\_\_1\_\_1\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_

\_\_1111\_\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_11\_1\_\_\_\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_1111\_\_\_\_\_\_\_\_1\_\_\_\_\_

\_11\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_1\_\_\_1\_\_\_\_\_\_\_\_1111\_\_\_\_\_\_11\_\_111\_\_\_\_\_\_\_\_\_\_\_\_

\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_

\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_1111\_\_\_\_\_\_\_\_\_

\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_

11\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_

1\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1111\_\_11\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_111\_\_11\_\_1\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_11

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_11\_1\_\_\_11\_11\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_\_11\_11\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_111111\_\_\_\_\_\_\_\_1\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_1\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_111\_\_11111\_\_\_\_\_1\_\_\_111\_\_\_\_\_\_\_\_\_\_11\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_1\_11111111\_1\_\_1\_\_\_11\_\_\_\_11111\_11\_\_\_\_\_\_\_\_\_1\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_11\_\_1\_\_1\_\_1\_\_1\_\_\_\_1111111\_\_1\_\_1\_\_\_\_\_\_\_\_\_1\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1\_111\_\_1\_\_1\_\_1\_\_\_\_\_1\_\_1\_11\_1\_\_\_\_\_\_\_\_\_\_\_\_1\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1111\_\_1\_11\_\_1\_\_\_\_\_1\_\_\_\_111\_11\_1\_\_\_1\_\_\_\_\_\_\_\_\_11\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_111\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_11\_1\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_111\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_\_111\_\_\_\_\_\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_111\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_11\_\_\_\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_11\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_1\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_1\_\_\_\_\_\_\_\_\_\_\_\_1111\_11\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_1\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_11111\_\_1\_\_\_\_\_\_\_11\_\_\_1\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_1\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_111\_\_\_\_\_1\_\_\_\_\_\_\_\_11\_\_1\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_1\_1\_\_\_\_\_1\_\_\_\_\_\_\_\_\_1111\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_11\_\_\_1\_\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_1\_\_11\_\_1\_\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_11\_\_1\_\_\_\_1\_\_\_\_\_11\_\_1\_\_\_1\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_\_11\_\_\_\_11\_\_\_1\_\_\_1\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1\_\_\_11111\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_1\_\_11\_\_1111111\_\_\_11\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_\_\_111\_\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_1\_1\_\_\_\_\_\_11\_\_\_\_\_\_\_\_1\_\_\_1\_\_1\_\_\_\_\_\_1\_\_\_\_\_\_111\_11\_\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_1\_\_\_1\_\_1\_\_\_\_\_\_1\_\_\_\_\_11\_\_\_\_11\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_1\_1\_\_\_\_\_\_\_\_111111\_\_1\_\_\_\_\_\_1\_\_\_\_11\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_11\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1111\_\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_1\_11\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_1\_1\_\_\_\_\_\_11\_\_\_1\_\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_1\_\_1\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_1\_\_\_\_\_\_\_\_\_11\_\_1\_\_\_\_\_\_\_1\_\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_1\_\_\_\_\_\_\_\_\_\_11111\_\_\_\_\_111\_\_

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111111\_\_\_\_\_\_\_\_\_11\_\_\_11\_\_\_\_1\_1\_\_

\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_1\_\_\_\_1\_\_\_\_

\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_111\_\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_11\_\_\_\_

1\_\_\_\_\_1\_\_1\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_11\_1\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_11\_\_\_\_\_

11\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_11\_\_\_\_\_

\_11\_\_\_11\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_1\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_11\_\_\_\_\_\_

\_\_1\_\_\_111\_\_\_\_1\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_1\_\_\_11\_1\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_111\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_111\_\_\_11\_\_\_\_\_\_\_

\_\_1\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_1\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1111\_\_\_\_\_\_\_\_\_\_\_11111\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_111\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_11\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_1\_\_\_1\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_