**Computer Vision HW6: Yokoi Connectivity Number**

**R10741015 鄭傑鴻**

**Oct. 21, 2022**

**Original Image:**

****

**Step 1: Down Sampling**

* Description:

Get the binary image by taking a threshold=128, and then take the topmost-left pixel in the 8x8 block.

* Algorithm:

*binary\_pic = original\_pic >= 128 (Thresholding, using the broadcast property)*

*for each pixel:*

*assign binary[i\*8][j\*8] to binary\_downsampled[i][j] (Down sampling)*

* Code:

self.binary = (self.pic >= 128).astype(int)

self.binary\_ds = np.zeros((64, 64)).astype(int)

self.downsample()

def downsample(self):  
 for i in range(64):  
 for j in range(64):  
 self.binary\_ds[i][j] = self.binary[i\*8][j\*8]  
 self.printImg(self.binary\_ds)

* Result:

Down sampled Binary Image (Value 0 replaced by ‘\_’ to better visualize)**:**

11111111\_\_\_\_\_\_\_\_11111111111111111111\_\_\_\_\_111111111111\_\_\_\_\_\_\_1\_\_1

11111111\_\_\_\_\_\_\_\_\_111111111111\_1\_11\_\_11\_\_\_1111111111111\_\_\_\_\_\_\_\_1\_

11111111\_\_\_\_\_\_\_\_1\_1111111111\_\_11111111\_\_\_\_111111111111\_\_\_\_\_\_11\_\_

11111111\_\_\_\_\_\_\_\_1\_1\_111111\_11111111\_\_\_\_\_\_\_1111111111111\_\_\_\_\_1\_\_\_

11111111\_\_\_\_\_\_\_\_\_11\_1111\_11\_\_\_\_111\_1\_1\_\_\_\_11111111111111\_\_\_1\_\_\_\_

11111111\_\_\_\_\_\_\_\_\_1\_\_1\_\_11\_1\_\_\_\_\_1\_\_\_1\_\_\_\_\_11111111111111\_\_1\_\_\_\_\_

11111111\_\_\_\_\_\_\_\_\_\_\_11\_1\_\_111111\_\_\_\_1111\_\_\_111111111111111\_\_\_\_\_\_\_

11111111\_\_\_\_\_\_\_\_\_\_\_1111\_1111111111\_\_\_\_\_\_\_\_111111111111111\_\_\_\_\_\_\_

111\_1111\_\_\_\_\_\_\_\_\_\_\_\_1\_\_111111111111\_\_\_\_\_\_\_111111111111111\_\_\_\_\_\_\_

11\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111111111\_\_\_\_\_\_11111111111111\_\_\_\_\_\_\_\_

11\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_11111111111\_\_\_\_1111\_11111111\_\_\_\_\_\_\_\_\_

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

\_\_\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1111111111111111\_\_\_1111\_\_\_11111\_\_\_\_\_\_\_\_11

\_\_\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111111111111111\_\_1111\_\_\_1111\_\_\_\_\_\_\_\_111

\_\_\_\_1111\_\_\_\_\_\_\_\_1\_\_\_\_\_1111111111111111111\_1111\_\_\_\_11\_\_\_\_\_\_\_\_1111

\_\_\_\_1111\_\_\_\_\_\_\_\_1\_\_\_\_11\_1\_1111111111111111\_111\_\_11111\_\_\_\_\_\_\_1111

\_\_\_\_1111\_\_\_\_\_\_\_\_1\_\_\_\_1\_\_\_11111111111111111\_111\_111111\_\_\_\_\_\_11111

\_\_\_\_1111\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_11111111111111111111111111111\_\_\_\_\_111111

\_\_\_\_1111\_\_\_\_\_\_\_11\_\_\_\_\_\_11111111111111111111111111111\_\_\_\_\_\_111111

\_\_\_\_1111\_\_\_\_\_\_\_11\_\_\_1\_111111111111111111111111111111\_\_\_\_\_1111111

\_\_\_\_1111\_\_\_\_\_\_\_111\_\_\_11\_11111111111111111111111111\_1\_\_\_\_\_1111111

\_\_\_\_1111\_\_\_\_\_\_\_1111\_\_1\_111111111111111111111111111\_\_\_\_\_\_11111111

\_\_\_\_1111\_\_\_\_\_\_\_11111\_\_1\_111\_1\_11\_\_1\_\_11111111111\_\_1\_\_\_\_\_11111111

\_\_\_\_1111\_\_\_\_\_\_\_1111\_\_111\_1\_\_\_\_\_\_\_\_\_\_1111111111\_\_\_1\_\_\_\_\_111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_111\_1\_\_111\_\_\_\_\_\_\_\_\_111111111\_\_111\_\_\_\_\_\_111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_1111\_\_\_1\_\_\_\_\_\_\_\_\_\_\_1111111\_\_\_111\_\_\_\_\_\_1111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_1\_\_1\_\_1\_\_\_\_\_\_\_\_\_\_111111111\_\_\_1\_\_\_\_\_\_\_1111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_1\_\_\_1\_\_\_\_\_\_1\_\_\_\_1111111111\_\_1\_\_\_\_\_\_\_1\_111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111111111\_\_\_\_\_\_\_\_\_\_11111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_1\_\_1\_\_\_\_\_\_\_\_\_\_111111111111\_\_\_\_\_\_\_\_\_\_11111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_11111111111111\_\_1\_\_\_\_\_111111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_1\_1\_\_\_\_\_\_\_\_\_\_11111\_\_1111111\_\_\_1\_\_\_\_\_111111111111

\_\_\_\_1111\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_11111\_\_\_\_1\_\_\_\_\_111111111111

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

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

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

\_\_\_\_1111\_\_\_\_\_1\_\_\_\_\_\_11\_\_\_\_11\_\_111111111111\_\_\_\_1\_\_\_11111111111111

\_\_\_\_1111\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_1111111111111\_\_\_1\_\_\_11111111111111

\_\_\_\_1111\_\_\_1\_\_\_\_1\_\_\_\_\_1\_\_11\_\_11111111\_11111\_\_\_\_1\_\_11111111111111

\_\_\_\_1111\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_1111111\_11111\_\_\_\_1\_111111111111111

\_\_\_\_1111\_\_\_\_\_\_1\_1\_\_\_\_\_\_11\_\_\_\_\_111111\_1\_111\_\_\_\_\_1\_111111111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_11111111\_111\_\_\_\_\_1\_111111111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_1\_\_\_1\_1\_\_\_\_\_1111111111111\_\_\_\_\_1\_111111111111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_1\_\_11\_\_\_\_\_\_\_111111111111\_\_\_\_\_\_11111111111111111

\_\_\_\_1111\_\_1\_\_\_\_\_\_1\_11\_\_\_\_\_\_\_\_\_\_111\_\_\_\_1\_1\_\_\_\_\_\_11111111111111111

\_\_\_\_1111\_\_\_\_\_\_\_1\_\_1\_\_1\_\_\_\_\_\_\_\_\_1111\_\_111\_\_\_\_\_\_\_111111111\_1111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_11111111\_\_\_\_\_\_\_111111111\_1111111

\_\_\_\_1111\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111111\_\_\_\_\_\_\_\_111111111\_1111111

\_\_\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111\_\_\_\_\_\_\_\_111111111\_111111\_

\_\_\_\_11111\_\_\_\_\_\_1\_\_\_11\_\_\_\_\_\_\_\_\_\_111111111\_\_\_\_\_\_\_1\_\_\_\_\_11\_111111\_\_

1\_\_\_\_111\_1\_\_\_\_\_1\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_111111111\_\_\_\_\_1111\_\_\_\_\_11111\_\_\_

11\_\_\_1111\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_11111111111\_\_\_111111\_\_\_1111\_\_\_\_

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

\_\_1\_\_111\_\_\_\_\_\_\_\_\_\_\_\_1\_1\_\_\_\_\_\_\_\_11111111111111\_\_111111\_\_1111\_\_\_\_\_

\_\_1\_\_1111\_\_\_\_1\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_111111111111111\_11111\_11111\_\_\_\_\_\_

\_\_1\_\_111\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_111111111111111\_111111\_1111\_\_\_\_\_\_

\_\_1\_\_1111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_111111111111111\_111111\_1111\_\_\_\_\_\_

\_\_11\_1111\_\_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_111111111111111\_\_111111111\_\_\_\_\_\_\_

\_\_11\_111\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_11111111111111111\_\_\_\_111111\_\_\_\_\_\_\_

\_\_11\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111111111111111\_\_\_\_\_\_111\_\_\_\_\_\_\_\_

\_\_11\_111\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_111111111111111111\_\_\_\_\_\_111\_\_\_\_\_\_\_\_

\_\_11\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1111111111111111111\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_

\_\_11\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_111111111111111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_11\_111\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_1111111111111111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 2: Counts the Yokoi Connectivity Number:**

* Description:

Using the formulas on the slide/textbook, set up function *h* to get the intermediate results, and use function *f* to get the final result. Apply both functions to all pixels.

* Algorithm:

*Function h:*

*Return ‘q’ if b==c and (d!=b or e!=b)*

*Return ‘r’ if b==c and (d==b and e==b)*

*Return ‘s’ if b!=c*

*Function f:*

*Return 5 if a1,a2,a3,a4 are all ‘r’*

*Else return the number of ‘q’s*

*For every pixel:*

*Calculate a1, a2, a3, a4 using the function h*

*Calculate the result using function f*

*Assign the result to answer[i][j]*

* Code:

def funH(self,b, c, d, e):  
 if b == c and (d!=b or e!=b): return 'q'  
 if b==c and (d==b and e==b): return 'r'  
 if b!=c: return 's'  
  
def funF(self, a1, a2, a3, a4):  
 if a1=='r' and a2=='r' and a3=='r' and a4=='r':  
 return '5'  
 else:  
 ret\_val = 0  
 if a1 == 'q': ret\_val += 1  
 if a2 == 'q': ret\_val += 1  
 if a3 == 'q': ret\_val += 1  
 if a4 == 'q': ret\_val += 1  
 return ret\_val  
  
def inRange(self, x, y):  
 if x >= 64 or x<0:  
 return False  
 if y >= 64 or y<0:  
 return False  
 return True

def yokoi(self):  
 for i in range(64):  
 for j in range(64):  
 if self.binary\_ds[i][j] == 1:  
 x0 = self.binary\_ds[i][j]  
 x1 = 0 if not self.inRange(i + 0, j + 1) else self.binary\_ds[i + 0][j + 1]  
 x2 = 0 if not self.inRange(i - 1, j + 0) else self.binary\_ds[i - 1][j + 0]  
 x3 = 0 if not self.inRange(i + 0, j - 1) else self.binary\_ds[i + 0][j - 1]  
 x4 = 0 if not self.inRange(i + 1, j + 0) else self.binary\_ds[i + 1][j + 0]  
 x5 = 0 if not self.inRange(i + 1, j + 1) else self.binary\_ds[i + 1][j + 1]  
 x6 = 0 if not self.inRange(i - 1, j + 1) else self.binary\_ds[i - 1][j + 1]  
 x7 = 0 if not self.inRange(i - 1, j - 1) else self.binary\_ds[i - 1][j - 1]  
 x8 = 0 if not self.inRange(i + 1, j - 1) else self.binary\_ds[i + 1][j - 1]  
  
 # Upper right: [0,0], [0, 1], [-1, 1], [-1, 0]  
 a1 = self.funH(x0, x1, x6, x2)  
 # Upper left: [0,0], [-1, 0], [-1, -1], [0, -1]  
 a2 = self.funH(x0, x2, x7, x3)  
 # Bottom left: [0,0], [0, -1], [1, -1], [1, 0]  
 a3 = self.funH(x0, x3, x8, x4)  
 # Bottom right: [0,0], [1, 0], [1, 1], [0, 1]  
 a4 = self.funH(x0, x4, x5, x1)  
 self.answer[i][j] = self.funF(a1, a2, a3, a4)  
 else: self.answer[i][j] = 0  
 self.printImg(self.answer)

* Result:

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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_