In [30]: f = open("US\_AIR\_DATA.txt") In [2]: list\_ = f.readlines() In [3]: u = [] V = []for i in list : getArray = i.split("\t") getU = getArray[0] getV = getArray[1][:-1] u.append(int(getU)-1) v.append(int(getV)-1) In [4]: nodes = max(max(u), max(v)) + 1print('The total number of Nodes are : ', nodes) print('The Number of edges are : ',len(u)) # AdjacencyMatrix is created. m = nodesn = nodesadjacencyMatrix = [[0 for x in range(m)] for y in range(n)]for i in range(len(u)): adjacencyMatrix[u[i]][v[i]] = 1adjacencyMatrix[v[i]][u[i]] = 1The total number of Nodes are: 332 The Number of edges are: 2126 In [5]: count = 0 for i in range(m): for j in range(n): if adjacencyMatrix[i][j] == 1: count = count + 1print('Number of 1''s present are : ',count) Number of 1s present are: 4252 In [6]: #Creating an Edge List #Edge List is created as an Dictionary where key is the Node and Value is a list consists of all the neighbours nodes edgeList = {} for i in range(nodes): getList = [] # j is for Columns for j in range(nodes): if adjacencyMatrix[i][j] == 1 : getList.append(j) edgeList[i] = getList print('Edge List of the given data is : \n',edgeList) Edge List of the given data is {0: [1, 3, 7], 1: [0, 3, 7], 2: [4, 7], 3: [0, 1, 7, 25, 46], 4: [2, 7], 5: [6, 7, 12], 6: [5, 7, 12], 7: [0, 1, 2, 3, 4, 5, 6, 12, 15, 22, 23, 25, 26, 27, 29, 33, 34, 35, 36, 37, 46, 64, 66, 111, 117, 143, 200, 247, 312], 8: [9, 10, 11, 12], 9: [8, 10, 11, 12], 10: [8, 9, 1 2], 11: [8, 9, 12], 12: [5, 6, 7, 8, 9, 10, 11, 13, 14, 16, 17, 18, 19, 20], 13: [12], 14: [1 2], 15: [7, 21], 16: [12, 18, 19], 17: [12], 18: [12, 16, 19], 19: [12, 16, 18], 20: [12], 2 1: [15, 25], 22: [7, 23], 23: [7, 22], 24: [25], 25: [3, 7, 21, 24, 28, 30, 32, 46], 26: [7], 27: [7, 34], 28: [25, 30, 32, 46], 29: [7, 33], 30: [25, 28, 31], 31: [30, 32, 329], 32: [25, 28, 31, 46], 33: [7, 29], 34: [7, 27, 35], 35: [7, 34, 37], 36: [7, 37], 37: [7, 35, 36], 38: [46, 64, 141], 39: [44, 45, 49, 54, 60, 61, 143], 40: [52, 66, 165], 41: [46], 42: [66], 43: [66], 44: [39, 45, 46, 49, 53, 54, 57, 58, 60, 61, 62, 64, 66, 82, 117, 141, 143, 165, 200], 45: [39, 44, 49, 54, 61, 62, 66, 143], 46: [3, 7, 25, 28, 32, 38, 41, 44, 47, 48, 53, 55, 57, 58, 59, 64, 66, 73, 74, 82, 85, 93, 108, 111, 117, 122, 130, 141, 143, 146, 149, 151, 165, 16 8, 171, 175, 176, 181, 182, 196, 200, 202, 218, 229, 231, 244, 245, 247, 252, 254, 257, 260, 262, 292, 310, 312, 315], 47: [46, 48, 64], 48: [46, 47, 55, 59], 49: [39, 44, 45, 54, 60, 6 1, 62, 66, 143, 165], 50: [66, 81, 117, 165], 51: [66], 52: [40, 66, 81, 143, 165], 53: [44, 46, 57, 58, 59, 64, 82], 54: [39, 44, 45, 49, 60, 61, 62, 143], 55: [46, 48, 57, 58, 59, 64] 56: [68, 69, 70, 117], 57: [44, 46, 53, 55, 58, 59, 64, 82], 58: [44, 46, 53, 55, 57, 59, 63, 64, 82, 143], 59: [46, 48, 53, 55, 57, 58, 63, 64], 60: [39, 44, 49, 54, 61, 62], 61: [39, 4 4, 45, 49, 54, 60, 62, 66, 143, 165], 62: [44, 45, 49, 54, 60, 61, 66, 143, 165], 63: [58, 5 9, 64], 64: [7, 38, 44, 46, 47, 53, 55, 57, 58, 59, 63, 65, 66, 73, 74, 82, 86, 107, 111, 11 5, 117, 141, 143, 150, 165, 168, 175, 182, 196, 200, 202, 218, 244, 245, 247, 252, 254, 257, 260, 262, 292], 65: [64, 86, 115, 150], 66: [7, 40, 42, 43, 44, 45, 46, 49, 50, 51, 52, 61, 6 2, 64, 70, 75, 77, 78, 81, 89, 93, 94, 98, 108, 110, 111, 117, 118, 119, 122, 127, 130, 132, 135, 143, 145, 146, 149, 151, 152, 158, 160, 161, 165, 166, 171, 173, 175, 176, 178, 181, 18 2, 188, 196, 200, 202, 216, 218, 229, 231, 232, 245, 247, 252, 254, 257, 260, 262, 273, 291, 292, 295, 298, 300, 306, 309, 310, 312], 67: [79, 95, 108, 117, 124, 146], 68: [56, 70, 72, 1 17], 69: [56, 111, 117], 70: [56, 66, 68, 72, 76, 89, 110, 111, 117], 71: [95, 117, 145, 146, 151, 161, 176], 72: [68, 70, 76, 89, 93, 117], 73: [46, 64, 74, 115, 200], 74: [46, 64, 73, 8 6, 107, 115, 150, 165, 200], 75: [66, 81, 97, 117, 141, 143], 76: [70, 72, 93, 117], 77: [66, 78, 117], 78: [66, 77, 117], 79: [67, 100, 108, 117, 145, 146, 151, 161, 175, 176], 80: [84, 117, 143, 165, 260], 81: [50, 52, 66, 75, 105, 117, 129, 139, 143, 165, 181], 82: [44, 46, 5 3, 57, 58, 64, 84, 85, 96, 103, 117, 143, 165, 200, 247], 83: [111, 117], 84: [80, 82, 96, 14 3, 260], 85: [46, 82, 143], 86: [64, 65, 74], 87: [117], 88: [173], 89: [66, 70, 72, 93, 111, 117, 151, 165, 181], 90: [91, 94, 108, 111, 117, 118, 145, 146, 149, 151, 161, 173, 175, 176, 178, 220, 229, 254], 91: [90, 94, 108, 111, 117, 118, 142, 145, 146, 149, 151, 161, 173, 175, 178, 181, 220, 229, 254, 298], 92: [99, 117, 151, 160], 93: [46, 66, 72, 76, 89, 98, 108, 11 1, 117, 130, 145, 146, 151, 158, 161, 165, 171, 175, 178, 181, 200, 216, 218, 229, 231, 247, 254, 257, 260, 262, 298, 300, 309, 310], 94: [66, 90, 91, 100, 108, 111, 117, 118, 130, 145, 146, 149, 151, 161, 173, 175, 176, 178, 220, 229, 254, 298], 95: [67, 71, 117, 146, 151, 161, 176], 96: [82, 84, 103], 97: [75, 143], 98: [66, 93, 111, 117, 146, 151, 160, 166, 175], 99: [92, 110, 111, 117, 136, 151, 160], 100: [79, 94, 111, 117, 128, 145, 151, 161, 173, 175, 17 8, 220, 229, 254], 101: [106, 117, 119, 181], 102: [114, 151], 103: [82, 96, 143], 104: [122, 166, 181], 105: [81, 117, 119, 139, 165, 181], 106: [101, 117], 107: [64, 74, 115, 200], 108: [46, 66, 67, 79, 90, 91, 93, 94, 111, 117, 118, 130, 143, 145, 146, 149, 151, 158, 160, 161, 165, 166, 169, 173, 175, 176, 178, 181, 200, 201, 202, 211, 216, 218, 220, 229, 231, 247, 25 4, 257, 260, 272, 292, 298, 300, 305, 306, 309, 310, 320], 109: [151, 161], 110: [66, 70, 99, 111, 117, 151, 160], 111: [7, 46, 64, 66, 69, 70, 83, 89, 90, 91, 93, 94, 98, 99, 100, 108, 1 10, 117, 118, 122, 124, 125, 130, 135, 136, 143, 145, 146, 148, 149, 151, 156, 158, 160, 161, 165, 166, 169, 171, 173, 175, 176, 178, 181, 188, 200, 201, 202, 211, 216, 218, 220, 229, 23 1, 247, 252, 254, 257, 260, 262, 291, 292, 295, 298, 300, 304, 305, 306, 309, 310], 112: [15 1, 173], 113: [117], 114: [102, 151], 115: [64, 65, 73, 74, 107, 150], 116: [151], 117: [7, 4 4, 46, 50, 56, 64, 66, 67, 68, 69, 70, 71, 72, 75, 76, 77, 78, 79, 80, 81, 82, 83, 87, 89, 9 0, 91, 92, 93, 94, 95, 98, 99, 100, 101, 105, 106, 108, 110, 111, 113, 118, 119, 124, 125, 12 6, 127, 128, 129, 130, 132, 133, 135, 136, 138, 139, 142, 143, 144, 145, 146, 147, 148, 149, 151, 152, 153, 154, 156, 157, 158, 160, 161, 162, 163, 165, 166, 167, 168, 171, 172, 173, 17 5, 176, 178, 180, 181, 182, 185, 188, 190, 191, 196, 197, 200, 201, 202, 203, 211, 215, 216, 217, 218, 220, 221, 224, 228, 229, 231, 232, 239, 245, 247, 248, 249, 252, 254, 255, 257, 25 9, 260, 262, 272, 273, 275, 283, 287, 291, 292, 295, 296, 298, 300, 304, 305, 306, 309, 310, 312, 320], 118: [66, 90, 91, 94, 108, 111, 117, 130, 145, 146, 149, 151, 161, 173, 175, 176, 178, 181, 216, 220, 229, 254, 260, 298, 300, 305, 309, 310, 320], 119: [66, 101, 105, 117, 12 9, 165, 181, 257], 120: [141], 121: [162], 122: [46, 66, 104, 111, 130, 145, 146, 151, 158, 1 62, 165, 166, 171, 178, 181, 216, 218, 254, 260, 272, 295, 298, 300, 301, 306, 309], 123: [13 200], 124: [67, 111, 117, 130, 146, 151, 161, 169, 173, 176, 178, 220, 229, 254, 298], 12 5: [111, 117, 136, 151, 160, 175, 181, 216], 126: [117, 136, 151, 254], 127: [66, 117, 132, 1 41, 151, 165, 181, 257, 260], 128: [100, 117, 151, 220, 254], 129: [81, 117, 119, 144, 147, 59, 165, 181, 257], 130: [46, 66, 93, 94, 108, 111, 117, 118, 122, 124, 145, 146, 149, 151, 1 52, 161, 165, 166, 169, 171, 173, 175, 176, 178, 181, 188, 200, 211, 216, 218, 220, 229, 247, 254, 257, 260, 291, 292, 298, 300, 304, 305, 306, 309, 310], 131: [151, 173, 229, 254], 132: [66, 117, 127, 139, 143, 151, 165, 171, 181, 218, 231, 254, 257, 260, 292], 133: [117, 135] 134: [138], 135: [66, 111, 117, 133, 151], 136: [99, 111, 117, 125, 126, 138, 151, 160, 175, 254], 137: [123, 182, 200], 138: [117, 134, 136, 160, 161, 166, 176, 229], 139: [81, 105, 11 7, 132, 165, 181], 140: [156, 176], 141: [38, 44, 46, 64, 75, 120, 127, 143, 165, 171, 197, 2 15, 224, 228, 232, 260, 292], 142: [91, 117, 146, 151, 161, 169, 173, 220, 229, 298, 305, 30 9, 310], 143: [7, 39, 44, 45, 46, 49, 52, 54, 58, 61, 62, 64, 66, 75, 80, 81, 82, 84, 85, 97, 103, 108, 111, 117, 132, 141, 149, 161, 165, 167, 168, 171, 175, 176, 180, 181, 182, 196, 20 0, 202, 212, 215, 218, 224, 232, 244, 245, 247, 249, 252, 254, 257, 260, 262, 273, 291, 292, 296, 298], 144: [117, 129, 147, 181], 145: [66, 71, 79, 90, 91, 93, 94, 100, 108, 111, 117, 1 18, 122, 130, 146, 149, 151, 158, 160, 165, 166, 167, 171, 173, 175, 176, 178, 181, 188, 201, 211, 216, 217, 220, 229, 231, 236, 254, 259, 260, 283, 291, 292, 298, 300, 304, 305, 306, 30 9, 310], 146: [46, 66, 67, 71, 79, 90, 91, 93, 94, 95, 98, 108, 111, 117, 118, 122, 124, 130 142, 145, 149, 151, 152, 158, 160, 161, 165, 166, 171, 173, 175, 176, 178, 181, 200, 201, 21 1, 216, 217, 218, 220, 229, 231, 247, 248, 254, 257, 259, 260, 272, 283, 291, 292, 297, 298, 299, 300, 304, 305, 306, 309, 310, 319, 320, 321, 323, 324], 147: [117, 129, 144, 157, 165, 1 81], 148: [111, 117, 151, 156, 220, 229, 254], 149: [46, 66, 90, 91, 94, 108, 111, 117, 118, 130, 143, 145, 146, 151, 158, 161, 165, 166, 173, 175, 176, 178, 181, 200, 201, 211, 218, 22 0, 229, 247, 254, 257, 260, 262, 291, 295, 298, 300, 305, 309, 310, 319, 320, 321, 323, 324], 150: [64, 65, 74, 115, 200], 151: [46, 66, 71, 79, 89, 90, 91, 92, 93, 94, 95, 98, 99, 100, 1 02, 108, 109, 110, 111, 112, 114, 116, 117, 118, 122, 124, 125, 126, 127, 128, 130, 131, 132, 135, 136, 142, 145, 146, 148, 149, 155, 156, 157, 158, 160, 161, 165, 166, 169, 171, 173, 17 5, 176, 178, 181, 182, 185, 186, 188, 190, 191, 197, 200, 201, 209, 211, 214, 216, 217, 218, 220, 221, 229, 231, 232, 247, 251, 254, 255, 257, 260, 262, 283, 287, 291, 292, 296, 298, 30 0, 304, 305, 306, 309, 310], 152: [66, 117, 130, 146, 165, 202, 247, 260, 292], 153: [117, 16 3, 170, 181], 154; [117, 170], 155; [151], 156; [111, 117, 140, 148, 151, 176, 220, 229, 25 4], 157: [117, 147, 151, 160, 170, 181], 158: [66, 93, 108, 111, 117, 122, 145, 146, 149, 15 1, 160, 161, 165, 166, 173, 175, 178, 181, 216, 218, 229, 247, 254, 257, 260, 292, 298, 300] 159: [129, 181], 160: [66, 92, 98, 99, 108, 110, 111, 117, 125, 136, 138, 145, 146, 151, 157, 158, 161, 166, 173, 178, 181, 188, 190, 229, 247, 254, 260, 298, 300], 161; [66, 71, 79, 90, 91, 93, 94, 95, 100, 108, 109, 111, 117, 118, 124, 130, 138, 142, 143, 146, 149, 151, 158, 16 0, 165, 166, 171, 173, 175, 176, 178, 181, 200, 201, 211, 216, 217, 218, 220, 229, 231, 236, 242, 247, 248, 251, 254, 257, 260, 272, 283, 287, 291, 292, 298, 300, 304, 305, 306, 309, 31 0, 320], 162: [117, 121, 122, 181], 163: [117, 153, 181], 164: [200], 165: [40, 44, 46, 49, 5 0, 52, 61, 62, 64, 66, 74, 80, 81, 82, 89, 93, 105, 108, 111, 117, 119, 122, 127, 129, 130, 1 32, 139, 141, 143, 145, 146, 147, 149, 151, 152, 158, 161, 166, 167, 168, 171, 172, 173, 174, 175, 176, 180, 181, 182, 183, 196, 197, 200, 202, 205, 212, 215, 216, 218, 224, 229, 231, 23 2, 241, 244, 245, 247, 249, 250, 252, 254, 257, 260, 262, 273, 275, 287, 291, 292, 295, 296, 298, 300, 310, 312], 166: [66, 98, 104, 108, 111, 117, 122, 130, 138, 145, 146, 149, 151, 15 8, 160, 161, 165, 171, 173, 175, 178, 181, 188, 190, 200, 216, 218, 229, 231, 247, 254, 257, 260, 291, 292, 298, 300, 301, 306, 309, 310], 167: [117, 143, 145, 165, 247, 257, 260], 168: [46, 64, 117, 143, 165, 182, 196, 200, 202, 212, 218, 245, 247, 257, 260, 262], 169: [108, 11 1, 124, 130, 142, 151], 170: [153, 154, 157], 171: [46, 66, 93, 111, 117, 122, 130, 132, 141, 143, 145, 146, 151, 161, 165, 166, 173, 175, 178, 181, 197, 200, 215, 216, 218, 224, 229, 23 1, 247, 254, 257, 260, 262, 292, 298], 172: [117, 165, 247, 260], 173: [66, 88, 90, 91, 94, 1 00, 108, 111, 112, 117, 118, 124, 130, 131, 142, 145, 146, 149, 151, 158, 160, 161, 165, 166, 171, 175, 176, 181, 200, 201, 203, 211, 216, 217, 218, 220, 229, 247, 254, 257, 260, 272, 28 3, 291, 292, 298, 300, 304, 305, 306, 309, 310, 320], 174: [165], 175: [46, 64, 66, 79, 90, 9 1, 93, 94, 98, 100, 108, 111, 117, 118, 125, 130, 136, 143, 145, 146, 149, 151, 158, 161, 16 5, 166, 171, 173, 178, 181, 188, 191, 200, 201, 211, 216, 217, 218, 220, 221, 229, 231, 232, 238, 247, 254, 255, 257, 260, 262, 270, 283, 291, 295, 298, 300, 304, 305, 306, 309, 310], 17 6: [46, 66, 71, 79, 90, 94, 95, 108, 111, 117, 118, 124, 130, 138, 140, 143, 145, 146, 149, 1 51, 156, 161, 165, 173, 178, 181, 189, 200, 201, 203, 211, 217, 229, 239, 247, 254, 257, 259 260, 262, 272, 291, 292, 298, 300, 309, 310, 320], 177: [200], 178: [66, 90, 91, 93, 94, 100, 108, 111, 117, 118, 122, 124, 130, 145, 146, 149, 151, 158, 160, 161, 166, 171, 175, 176, 18 1, 188, 216, 217, 220, 229, 231, 254, 259, 260, 291, 292, 298, 300, 305, 306, 309, 310], 179: [181], 180: [117, 143, 165, 181, 187, 218, 257, 260], 181: [46, 66, 81, 89, 91, 93, 101, 104 105, 108, 111, 117, 118, 119, 122, 125, 127, 129, 130, 132, 139, 143, 144, 145, 146, 147, 14 9, 151, 153, 157, 158, 159, 160, 161, 162, 163, 165, 166, 171, 173, 175, 176, 178, 179, 180, 188, 190, 191, 194, 195, 197, 200, 202, 205, 206, 208, 210, 211, 215, 216, 218, 219, 220, 22 1, 224, 225, 229, 231, 232, 238, 245, 247, 249, 252, 254, 255, 257, 260, 262, 273, 283, 291, 292, 295, 296, 298, 300, 304, 305, 306, 309, 310, 312, 320], 182: [46, 64, 66, 117, 137, 143, 151, 165, 168, 196, 200, 202, 212, 218, 223, 244, 245, 247, 252, 257, 260, 262, 292], 183: [1 65, 254, 260, 292], 184: [200], 185: [117, 151, 191, 229], 186: [151], 187: [180], 188: [66, 111, 117, 130, 145, 151, 160, 166, 175, 178, 181, 191, 216, 229, 231, 254, 255, 260], 189: [1 76, 201, 229], 190: [117, 151, 160, 166, 181], 191: [117, 151, 175, 181, 185, 188, 203, 216, 221, 254], 192: [200], 193: [220], 194: [181, 206], 195: [181], 196: [46, 64, 66, 117, 143, 1 65, 168, 182, 200, 212, 218, 244, 245, 247, 249, 250, 252, 257, 260, 262], 197: [117, 141, 15 1, 165, 171, 181, 205, 215, 218, 231, 257, 260], 198: [200, 212], 199: [212], 200: [7, 44, 4 6, 64, 66, 73, 74, 82, 93, 107, 108, 111, 117, 123, 130, 137, 143, 146, 149, 150, 151, 161, 1 64, 165, 166, 168, 171, 173, 175, 176, 177, 181, 182, 184, 192, 196, 198, 202, 204, 212, 213, 216, 218, 223, 227, 229, 231, 232, 235, 241, 244, 245, 247, 249, 250, 252, 254, 257, 260, 26 2, 275, 291, 292, 298, 310, 312, 315, 317], 201: [108, 111, 117, 145, 146, 149, 151, 161, 17 3, 175, 176, 189, 203, 211, 216, 220, 229, 254, 260], 202: [46, 64, 66, 108, 111, 117, 143, 52, 165, 168, 181, 182, 200, 218, 244, 245, 247, 249, 250, 252, 257, 260, 262, 287, 292, 31 0], 203: [117, 173, 176, 191, 201, 217, 229], 204: [200, 212], 205: [165, 181, 197, 208, 238, 260], 206: [181, 194], 207: [257], 208: [181, 205], 209: [151], 210: [181], 211: [108, 111, 1 17, 130, 145, 146, 149, 151, 161, 173, 175, 176, 181, 201, 216, 220, 229, 254, 260], 212: [14 3, 165, 168, 182, 196, 198, 199, 200, 204, 244, 245, 247, 252, 260, 262], 213: [200, 247], 21 4: [151, 229, 254], 215: [117, 141, 143, 165, 171, 181, 197, 224, 231, 238, 254, 257, 260, 26 1, 292, 295], 216: [66, 93, 108, 111, 117, 118, 122, 125, 130, 145, 146, 151, 158, 161, 165, 166, 171, 173, 175, 178, 181, 188, 191, 200, 201, 211, 217, 218, 220, 221, 229, 231, 238, 23 9, 247, 254, 255, 257, 260, 262, 270, 280, 282, 283, 285, 291, 292, 295, 296, 298, 300, 305, 306, 309, 310, 320], 217: [117, 145, 146, 151, 161, 173, 175, 176, 178, 203, 216, 220, 229, 2 34, 248, 254, 260, 298], 218: [46, 64, 66, 93, 108, 111, 117, 122, 130, 132, 143, 146, 149, 1 51, 158, 161, 165, 166, 168, 171, 173, 175, 180, 181, 182, 196, 197, 200, 202, 216, 229, 231, 232, 244, 245, 247, 249, 252, 254, 257, 260, 262, 273, 275, 292, 296, 298, 300, 312], 219: [1 81, 225, 260], 220: [90, 91, 94, 100, 108, 111, 117, 118, 124, 128, 130, 142, 145, 146, 148, 149, 151, 156, 161, 173, 175, 178, 181, 193, 201, 211, 216, 217, 229, 236, 242, 248, 251, 25 4, 259, 260, 272, 283, 297, 298, 299, 300, 304, 305, 306, 309, 310, 320, 321, 324], 221: [11 7, 151, 175, 181, 191, 216, 229, 231, 254, 260], 222: [229, 254], 223: [182, 200, 241, 260], 224: [117, 141, 143, 165, 171, 181, 215, 231, 254, 257, 260, 261, 292, 295], 225: [181, 219, 260], 226: [229], 227: [200, 235], 228: [117, 141, 232, 253, 260, 261, 273], 229: [46, 66, 9 0, 91, 93, 94, 100, 108, 111, 117, 118, 124, 130, 131, 138, 142, 145, 146, 148, 149, 151, 15 6, 158, 160, 161, 165, 166, 171, 173, 175, 176, 178, 181, 185, 188, 189, 200, 201, 203, 211, 214, 216, 217, 218, 220, 221, 222, 226, 231, 233, 234, 236, 237, 238, 239, 242, 247, 248, 25 1, 254, 255, 257, 258, 259, 260, 262, 271, 272, 280, 283, 285, 286, 287, 291, 292, 294, 296, 297, 298, 299, 300, 304, 305, 306, 309, 310, 320], 230: [257], 231: [46, 66, 93, 108, 111, 11 7, 132, 145, 146, 151, 161, 165, 166, 171, 175, 178, 181, 188, 197, 200, 215, 216, 218, 221, 224, 229, 238, 239, 247, 254, 255, 257, 260, 282, 284, 287, 291, 295, 296, 298, 300, 306, 30 9, 310], 232: [66, 117, 141, 143, 151, 165, 175, 181, 200, 218, 228, 247, 253, 254, 257, 260, 261, 262, 273, 274, 275, 292, 295], 233: [229, 254], 234: [217, 229], 235: [200, 227], 236: [145, 161, 220, 229, 254], 237: [229], 238: [175, 181, 205, 215, 216, 229, 231, 254, 260, 26 1], 239: [117, 176, 216, 229, 231, 254, 255, 260], 240: [260], 241: [165, 200, 223, 244, 247, 250, 260], 242: [161, 220, 229], 243: [247], 244: [46, 64, 143, 165, 182, 196, 200, 202, 212, 218, 241, 245, 247, 252, 257, 260, 262], 245: [46, 64, 66, 117, 143, 165, 168, 181, 182, 196, 200, 202, 212, 218, 244, 249, 252, 254, 257, 260, 262, 292, 312], 246: [260], 247: [7, 46, 6 4, 66, 82, 93, 108, 111, 117, 130, 143, 146, 149, 151, 152, 158, 160, 161, 165, 166, 167, 16 8, 171, 172, 173, 175, 176, 181, 182, 196, 200, 202, 212, 213, 216, 218, 229, 231, 232, 241, 243, 244, 249, 252, 254, 257, 260, 262, 273, 275, 291, 292, 298, 300, 310, 312, 315, 317, 33 0], 248: [117, 146, 161, 217, 220, 229, 254, 258, 259], 249: [117, 143, 165, 181, 196, 200, 2 02, 218, 245, 247, 257, 260, 262, 273], 250: [165, 196, 200, 202, 241, 257, 260], 251: [151, 161, 220, 229], 252: [46, 64, 66, 111, 117, 143, 165, 181, 182, 196, 200, 202, 212, 218, 244 245, 247, 254, 257, 260, 262, 292], 253: [228, 232, 260, 261, 275, 287], 254: [46, 64, 66, 9 0, 91, 93, 94, 100, 108, 111, 117, 118, 122, 124, 126, 128, 130, 131, 132, 136, 143, 145, 14 6, 148, 149, 151, 156, 158, 160, 161, 165, 166, 171, 173, 175, 176, 178, 181, 183, 188, 191, 200, 201, 211, 214, 215, 216, 217, 218, 220, 221, 222, 224, 229, 231, 232, 233, 236, 238, 23 9, 245, 247, 248, 252, 255, 257, 258, 259, 260, 262, 264, 265, 266, 270, 271, 272, 275, 280, 282, 283, 285, 286, 287, 288, 291, 292, 294, 295, 296, 297, 298, 299, 300, 304, 305, 306, 30 9, 310, 312, 320, 321], 255: [117, 151, 175, 181, 188, 216, 229, 231, 239, 254, 260, 266, 27 0, 271, 291, 295, 298], 256: [260], 257: [46, 64, 66, 93, 108, 111, 117, 119, 127, 129, 130, 132, 143, 146, 149, 151, 158, 161, 165, 166, 167, 168, 171, 173, 175, 176, 180, 181, 182, 19 6, 197, 200, 202, 207, 215, 216, 218, 224, 229, 230, 231, 232, 244, 245, 247, 249, 250, 252, 254, 260, 262, 263, 273, 275, 287, 292, 295, 296, 298, 312], 258: [229, 248, 254, 272], 259: [117, 145, 146, 176, 178, 220, 229, 248, 254, 272, 283], 260: [46, 64, 66, 80, 84, 93, 108, 1 11, 117, 118, 122, 127, 130, 132, 141, 143, 145, 146, 149, 151, 152, 158, 160, 161, 165, 166, 167, 168, 171, 172, 173, 175, 176, 178, 180, 181, 182, 183, 188, 196, 197, 200, 201, 202, 20 5, 211, 212, 215, 216, 217, 218, 219, 220, 221, 223, 224, 225, 228, 229, 231, 232, 238, 239, 240, 241, 244, 245, 246, 247, 249, 250, 252, 253, 254, 255, 256, 257, 262, 265, 266, 267, 26 8, 269, 270, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 285, 287, 289, 290, 291, 292, 293, 295, 296, 298, 300, 302, 303, 304, 305, 306, 307, 308, 309, 310, 312, 320], 26 1: [215, 224, 228, 232, 238, 253, 274, 275, 287, 291, 292, 295, 296], 262: [46, 64, 66, 93, 1 11, 117, 143, 149, 151, 165, 168, 171, 175, 176, 181, 182, 196, 200, 202, 212, 216, 218, 229 232, 244, 245, 247, 249, 252, 254, 257, 260, 275, 292, 310, 312], 263: [257], 264: [254], 26 5: [254, 260, 266, 270, 271, 280, 285], 266: [254, 255, 260, 265, 270, 280, 282], 267: [260], 268: [260, 269], 269: [260, 268], 270: [175, 216, 254, 255, 260, 265, 266, 280, 282], 271: [2 29, 254, 255, 265, 282], 272: [108, 117, 122, 146, 161, 173, 176, 220, 229, 254, 258, 259, 26 0, 283, 298, 321, 324], 273: [66, 117, 143, 165, 181, 218, 228, 232, 247, 249, 257, 260, 29 2], 274: [232, 260, 261, 275, 287, 295], 275: [117, 165, 200, 218, 232, 247, 253, 254, 257, 2 60, 261, 262, 274, 287, 292, 295, 296], 276: [260], 277: [260], 278: [260], 279: [260], 280: [216, 229, 254, 260, 265, 266, 270, 282, 285, 292], 281: [260], 282: [216, 231, 254, 260, 26 6, 270, 271, 280, 292], 283: [117, 145, 146, 151, 161, 173, 175, 181, 216, 220, 229, 254, 25 9, 260, 272, 285, 292, 297, 298, 299, 305, 310], 284: [231], 285: [216, 229, 254, 260, 265, 2 231, 253, 254, 257, 260, 261, 274, 275, 292, 295, 296, 307], 288: [254], 289: [260, 292], 29 0: [260], 291: [66, 111, 117, 130, 143, 145, 146, 149, 151, 161, 165, 166, 173, 175, 176, 17 8, 181, 200, 216, 229, 231, 247, 254, 255, 260, 261, 292, 295, 296, 298, 300, 309, 310], 292: [46, 64, 66, 108, 111, 117, 130, 132, 141, 143, 145, 146, 151, 152, 158, 161, 165, 166, 171, 173, 176, 178, 181, 182, 183, 200, 202, 215, 216, 218, 224, 229, 232, 245, 247, 252, 254, 25 7, 260, 261, 262, 273, 275, 280, 282, 283, 285, 287, 289, 291, 295, 296, 298, 300, 302, 304, 305, 306, 307, 308, 309, 310], 293: [260], 294: [229, 254], 295: [66, 111, 117, 122, 149, 16 5, 175, 181, 215, 216, 224, 231, 232, 254, 255, 257, 260, 261, 274, 275, 287, 291, 292, 296, 302, 307, 310], 296: [117, 143, 151, 165, 181, 216, 218, 229, 231, 254, 257, 260, 261, 275, 2 87, 291, 292, 295, 307], 297: [146, 220, 229, 254, 283, 298, 299], 298: [66, 91, 93, 94, 108, 111, 117, 118, 122, 124, 130, 142, 143, 145, 146, 149, 151, 158, 160, 161, 165, 166, 171, 17 3, 175, 176, 178, 181, 200, 216, 217, 218, 220, 229, 231, 247, 254, 255, 257, 260, 272, 283, 285, 286, 291, 292, 297, 300, 304, 305, 306, 309, 310, 320, 321, 324], 299: [146, 220, 229, 2 54, 283, 297, 300, 306], 300: [66, 93, 108, 111, 117, 118, 122, 130, 145, 146, 149, 151, 158, 160, 161, 165, 166, 173, 175, 176, 178, 181, 216, 218, 220, 229, 231, 247, 254, 260, 285, 28 6, 291, 292, 298, 299, 304, 305, 306, 309, 310, 320], 301: [122, 166, 309], 302: [260, 292, 2 95], 303: [260], 304: [111, 117, 130, 145, 146, 151, 161, 173, 175, 181, 220, 229, 254, 260, 292, 298, 300, 306, 310], 305: [108, 111, 117, 118, 130, 142, 145, 146, 149, 151, 161, 173, 1 75, 178, 181, 216, 220, 229, 254, 260, 283, 292, 298, 300, 306, 309, 310], 306: [66, 108, 11 1, 117, 122, 130, 145, 146, 151, 161, 166, 173, 175, 178, 181, 216, 220, 229, 231, 254, 260, 292, 298, 299, 300, 304, 305, 309, 310], 307: [260, 287, 292, 295, 296], 308: [260, 292], 30 9: [66, 93, 108, 111, 117, 118, 122, 130, 142, 145, 146, 149, 151, 161, 166, 173, 175, 176, 1 78, 181, 216, 220, 229, 231, 254, 260, 291, 292, 298, 300, 301, 305, 306, 310], 310: [46, 66, 93, 108, 111, 117, 118, 130, 142, 145, 146, 149, 151, 161, 165, 166, 173, 175, 176, 178, 181, 200, 202, 216, 220, 229, 231, 247, 254, 260, 262, 283, 286, 291, 292, 295, 298, 300, 304, 30 5, 306, 309, 319, 320, 321, 323, 324], 311: [312, 315, 317], 312: [7, 46, 66, 117, 165, 181, 200, 218, 245, 247, 254, 257, 260, 262, 311, 313, 314, 315, 316, 317, 318, 325, 328, 330], 31 3: [312, 315, 316], 314: [312, 315, 316], 315: [46, 200, 247, 311, 312, 313, 314, 316, 317, 3 18], 316: [312, 313, 314, 315], 317: [200, 247, 311, 312, 315, 318], 318: [312, 315, 317], 31 9: [146, 149, 310, 323], 320: [108, 117, 118, 146, 149, 161, 173, 176, 181, 216, 220, 229, 25 4, 260, 298, 300, 310, 321, 322, 323, 324], 321: [146, 149, 220, 254, 272, 298, 310, 320, 32 4], 322: [320], 323: [146, 149, 310, 319, 320], 324: [146, 149, 220, 272, 298, 310, 320, 32 1], 325: [312], 326: [327, 328, 329, 331], 327: [326, 328], 328: [312, 326, 327], 329: [31, 3 26], 330: [247, 312], 331: [326]} Visualizing the network Note: This is also done in Cytoscape. Cytoscape visualization is in report file In [7]: import networkx as nx import matplotlib.pvplot as plt import numpy as np In [8]: G = nx.from\_numpy\_matrix(np.array(adjacencyMatrix)) nx.draw(G, with\_labels=True) C:\Users\Lakshmi Praffulla\Anaconda3\lib\site-packages\networkx\drawing\nx\_pylab.py:579: Matp lotlibDeprecationWarning: The iterable function was deprecated in Matplotlib 3.1 and will be removed in 3.3. Use np.ite rable instead. if not cb.iterable(width): **Sparsness of the network** In [9]: #Sparsness of the network edgepossible = int((nodes\*(nodes-1))/2) edgespresent = len(u)print('Total edges possible for ', nodes, ' nodes are : ', edgepossible ) print('Number of edges present : ',edgespresent) print('Density of the graph is : ',edgespresent/edgepossible) Total edges possible for 332 nodes are: 54946 Number of edges present : 2126 Density of the graph is: 0.0386925344884068 **Average Degree** In [10]: # Computing its Average Degree nodeDegree = {} **for** i **in** edgeList: nodeDegree[i] = len(edgeList[i]) print('Degree of each node is : ') print(nodeDegree) sumOfDegree = 0for i in nodeDegree: sumOfDegree += nodeDegree[i] print() print('Average Degree of the given graph is : ',sumOfDegree/nodes) Degree of each node is: {0: 3, 1: 3, 2: 2, 3: 5, 4: 2, 5: 3, 6: 3, 7: 29, 8: 4, 9: 4, 10: 3, 11: 3, 12: 14, 13: 1, 1 4: 1, 15: 2, 16: 3, 17: 1, 18: 3, 19: 3, 20: 1, 21: 2, 22: 2, 23: 2, 24: 1, 25: 8, 26: 1, 27: 2, 28: 4, 29: 2, 30: 3, 31: 3, 32: 4, 33: 2, 34: 3, 35: 3, 36: 2, 37: 3, 38: 3, 39: 7, 40: 3, 41: 1, 42: 1, 43: 1, 44: 19, 45: 8, 46: 57, 47: 3, 48: 4, 49: 10, 50: 4, 51: 1, 52: 5, 53: 7, 54: 8, 55: 6, 56: 4, 57: 8, 58: 10, 59: 8, 60: 6, 61: 10, 62: 9, 63: 3, 64: 41, 65: 4, 66: 7 8, 67: 6, 68: 4, 69: 3, 70: 9, 71: 7, 72: 6, 73: 5, 74: 9, 75: 6, 76: 4, 77: 3, 78: 3, 79: 1 0, 80: 5, 81: 11, 82: 15, 83: 2, 84: 5, 85: 3, 86: 3, 87: 1, 88: 1, 89: 9, 90: 18, 91: 20, 9 2: 4, 93: 34, 94: 22, 95: 7, 96: 3, 97: 2, 98: 9, 99: 7, 100: 14, 101: 4, 102: 2, 103: 3, 10 4: 3, 105: 6, 106: 2, 107: 4, 108: 50, 109: 2, 110: 7, 111: 70, 112: 2, 113: 1, 114: 2, 115: 6, 116: 1, 117: 139, 118: 29, 119: 8, 120: 1, 121: 1, 122: 26, 123: 2, 124: 15, 125: 8, 126: 4, 127: 9, 128: 5, 129: 9, 130: 45, 131: 4, 132: 15, 133: 2, 134: 1, 135: 5, 136: 10, 137: 3, 138: 8, 139: 6, 140: 2, 141: 17, 142: 13, 143: 59, 144: 4, 145: 50, 146: 67, 147: 6, 148: 7, 149: 46, 150: 5, 151: 94, 152: 9, 153: 4, 154: 2, 155: 1, 156: 9, 157: 6, 158: 28, 159: 2, 16 0: 29, 161: 62, 162: 4, 163: 3, 164: 1, 165: 85, 166: 41, 167: 7, 168: 16, 169: 6, 170: 3, 17 1: 35, 172: 4, 173: 53, 174: 1, 175: 61, 176: 48, 177: 1, 178: 42, 179: 1, 180: 8, 181: 94, 1 82: 23, 183: 4, 184: 1, 185: 4, 186: 1, 187: 1, 188: 18, 189: 3, 190: 5, 191: 10, 192: 1, 19 3: 1, 194: 2, 195: 1, 196: 20, 197: 12, 198: 2, 199: 1, 200: 68, 201: 19, 202: 26, 203: 7, 20 4: 2, 205: 6, 206: 2, 207: 1, 208: 2, 209: 1, 210: 1, 211: 19, 212: 15, 213: 2, 214: 3, 215: 16, 216: 56, 217: 18, 218: 49, 219: 3, 220: 50, 221: 10, 222: 2, 223: 4, 224: 14, 225: 3, 22 6: 1, 227: 2, 228: 7, 229: 87, 230: 1, 231: 44, 232: 23, 233: 2, 234: 2, 235: 2, 236: 5, 237: 1, 238: 10, 239: 8, 240: 1, 241: 7, 242: 3, 243: 1, 244: 17, 245: 23, 246: 1, 247: 59, 248: 9, 249: 14, 250: 7, 251: 4, 252: 22, 253: 6, 254: 101, 255: 17, 256: 1, 257: 60, 258: 4, 259: 11, 260: 118, 261: 13, 262: 36, 263: 1, 264: 1, 265: 7, 266: 7, 267: 1, 268: 2, 269: 2, 270: 9, 271: 5, 272: 17, 273: 13, 274: 6, 275: 17, 276: 1, 277: 1, 278: 1, 279: 1, 280: 10, 281: 1, 282: 9, 283: 22, 284: 1, 285: 10, 286: 5, 287: 18, 288: 1, 289: 2, 290: 1, 291: 33, 292: 6 2, 293: 1, 294: 2, 295: 27, 296: 19, 297: 7, 298: 56, 299: 8, 300: 42, 301: 3, 302: 3, 303: 1, 304: 19, 305: 27, 306: 29, 307: 5, 308: 2, 309: 34, 310: 47, 311: 3, 312: 24, 313: 3, 314: 3, 315: 10, 316: 4, 317: 6, 318: 3, 319: 4, 320: 21, 321: 9, 322: 1, 323: 5, 324: 8, 325: 1, 326: 4, 327: 2, 328: 3, 329: 2, 330: 2, 331: 1} Average Degree of the given graph is: 12.80722891566265 'scaled degree distribution' In [11]: # Plot Scaled Degree Distribution maxDegree = 0for i in nodeDegree: if maxDegree < nodeDegree[i]:</pre> maxDegree = nodeDegree[i] degreeDistribution = {} for i in range(maxDegree+1): **if** i == 0: continue; degreeDistribution[i] = [] **for** i **in** nodeDegree: degreeDistribution[nodeDegree[i]].append(i) print('Degree Distribution for each node is : ',degreeDistribution) Degree Distribution for each node is : {1: [13, 14, 17, 20, 24, 26, 41, 42, 43, 51, 87, 88, 113, 116, 120, 121, 134, 155, 164, 174, 177, 179, 184, 186, 187, 192, 193, 195, 199, 207, 20 9, 210, 226, 230, 237, 240, 243, 246, 256, 263, 264, 267, 276, 277, 278, 279, 281, 284, 288, 290, 293, 303, 322, 325, 331], 2: [2, 4, 15, 21, 22, 23, 27, 29, 33, 36, 83, 97, 102, 106, 10 9, 112, 114, 123, 133, 140, 154, 159, 194, 198, 204, 206, 208, 213, 222, 227, 233, 234, 235, 268, 269, 289, 294, 308, 327, 329, 330], 3: [0, 1, 5, 6, 10, 11, 16, 18, 19, 30, 31, 34, 35, 37, 38, 40, 47, 63, 69, 77, 78, 85, 86, 96, 103, 104, 137, 163, 170, 189, 214, 219, 225, 242, 301, 302, 311, 313, 314, 318, 328], 4: [8, 9, 28, 32, 48, 50, 56, 65, 68, 76, 92, 101, 107, 1 26, 131, 144, 153, 162, 172, 183, 185, 223, 251, 258, 316, 319, 326], 5: [3, 52, 73, 80, 84, 128, 135, 150, 190, 236, 271, 286, 307, 323], 6: [55, 60, 67, 72, 75, 105, 115, 139, 147, 15 7, 169, 205, 253, 274, 317], 7: [39, 53, 71, 95, 99, 110, 148, 167, 203, 228, 241, 250, 265, 266, 297], 8: [25, 45, 54, 57, 59, 119, 125, 138, 180, 239, 299, 324], 9: [62, 70, 74, 89, 9 8, 127, 129, 152, 156, 248, 270, 282, 321], 10: [49, 58, 61, 79, 136, 191, 221, 238, 280, 28 5, 315], 11: [81, 259], 12: [197], 13: [142, 261, 273], 14: [12, 100, 224, 249], 15: [82, 12 4, 132, 212], 16: [168, 215], 17: [141, 244, 255, 272, 275], 18: [90, 188, 217, 287], 19: [4 4, 201, 211, 296, 304], 20: [91, 196], 21: [320], 22: [94, 252, 283], 23: [182, 232, 245], 2 4: [312], 25: [], 26: [122, 202], 27: [295, 305], 28: [158], 29: [7, 118, 160, 306], 30: [], 31: [], 32: [], 33: [291], 34: [93, 309], 35: [171], 36: [262], 37: [], 38: [], 39: [], 40: [], 41: [64, 166], 42: [178, 300], 43: [], 44: [231], 45: [130], 46: [149], 47: [310], 48: [1 76], 49: [218], 50: [108, 145, 220], 51: [], 52: [], 53: [173], 54: [], 55: [], 56: [216, 29 8], 57: [46], 58: [], 59: [143, 247], 60: [257], 61: [175], 62: [161, 292], 63: [], 64: [], 6 5: [], 66: [], 67: [146], 68: [200], 69: [], 70: [111], 71: [], 72: [], 73: [], 74: [], 75: [], 76: [], 77: [], 78: [66], 79: [], 80: [], 81: [], 82: [], 83: [], 84: [], 85: [165], 86: [], 87: [229], 88: [], 89: [], 90: [], 91: [], 92: [], 93: [], 94: [151, 181], 95: [], 96: [], 97: [], 98: [], 99: [], 100: [], 101: [254], 102: [], 103: [], 104: [], 105: [], 106: [], 107: [], 108: [], 109: [], 110: [], 111: [], 112: [], 113: [], 114: [], 115: [], 116: [], 11 7: [], 118: [260], 119: [], 120: [], 121: [], 122: [], 123: [], 124: [], 125: [], 126: [], 12 7: [], 128: [], 129: [], 130: [], 131: [], 132: [], 133: [], 134: [], 135: [], 136: [], 137: [], 138: [], 139: [117]} In [12]: #Plotting the graphs for different distributions def plotGraph(xValues, yValues, xLabel, yLabel, graphTitle, defaultSize='n'): if defaultSize=='n': f, ax=plt.subplots(figsize=(20,5)) ax.legend(fontsize = 14)plt.bar(xValues, yValues, color='green') plt.xlabel(xLabel) plt.ylabel(yLabel) plt.title(graphTitle) plt.show() In [14]: max\_degree = 0 **for** i **in** degreeDistribution: if max\_degree < len(degreeDistribution[i]):</pre> max\_degree = len(degreeDistribution[i]) In [17]: # Graph Plot Xval = []Yval = []for i in degreeDistribution: Xval.append(i) Yval.append(len(degreeDistribution[i])/max\_degree) plotGraph(Xval, Yval, 'degree (k)', 'pk', 'Scaled degree distribution') No handles with labels found to put in legend. Scaled degree distribution 0.4 0.2 Average Path Length, Diameter and Average Clustering Coefficient. In [18]: # Average Path Length = Average of the shortest path between all the pair of nodes edgeListForBFS = [] for i in edgeList: edgeListForBFS.append(edgeList[i]) In [19]: def BFS(adj, src, dest, v, pred, dist): # a queue to maintain queue of vertices whose # adjacency list is to be scanned as per normal # DFS algorithm queue = [] # boolean array visited[] which stores the # information whether ith vertex is reached # at least once in the Breadth first search visited = [False for i in range(v)]; # initially all vertices are unvisited # so v[i] for all i is false # and as no path is yet constructed # dist[i] for all i set to infinity for i in range(v): dist[i] = 1000000pred[i] = -1;# now source is first to be visited and # distance from source to itself should be 0 visited[src] = True; dist[src] = 0;queue.append(src); # standard BFS algorithm while (len(queue) != 0): u = queue[0];queue.pop(0); for i in range(len(adj[u])): if (visited[adj[u][i]] == False): visited[adj[u][i]] = True; dist[adj[u][i]] = dist[u] + 1;pred[adj[u][i]] = u;queue.append(adj[u][i]); # We stop BFS when we find # destination. **if** (adj[u][i] == dest): return True; return False; In [21]: def getShortestDistance(adj, s, dest, v): # predecessor[i] array stores predecessor of # i and distance array stores distance of i pred=[0 for i in range(v)] dist=[0 for i in range(v)]; if (BFS(adj, s, dest, v, pred, dist) == False): print("Given source and destination are not connected") # vector path stores the shortest path path = []crawl = dest;crawl = dest; path.append(crawl); while (pred[crawl] != -1): path.append(pred[crawl]); crawl = pred[crawl]; return dist[dest] In [22]: ShortestPathBetweenNodes = {} for i in range(len(edgeList)): for j in range(i+1,len(edgeList)): ShortestPathBetweenNodes[(i,j)] = getShortestDistance(edgeListForBFS, i, j, nodes)In [31]: # ShortestPathBetweenNodes In [24]: #Average Path Length ShortestPathSum = 0for i in ShortestPathBetweenNodes: ShortestPathSum += ShortestPathBetweenNodes[i] print('Average Shortest Path is ',ShortestPathSum/len(ShortestPathBetweenNodes)) Average Shortest Path is 2.742110435700506 In [26]: #Diameter - It is the longest shortest path between 2 nodes **for** i in ShortestPathBetweenNodes: if max < ShortestPathBetweenNodes[i]:</pre> max\_ = ShortestPathBetweenNodes[i] print('Longest Shortest Path is : ',max\_) print('Diameter nodes are : ') for i in ShortestPathBetweenNodes: if ShortestPathBetweenNodes[i] == max\_: print(i) Longest Shortest Path is: 6 Diameter nodes are : (8, 329)(8, 331)(9, 329)(9, 331)(10, 329)(10, 331)(11, 329)(11, 331)(13, 329)(13, 331)(14, 329)(14, 331)(16, 329)(16, 331)(17, 329)(17, 331)(18, 329)(18, 331)(19, 329)(19, 331)(20, 329)(20, 331)(21, 331)(24, 331)(39, 331)(54, 331)(60, 331)(63, 331)(65, 331)(86, 331)(88, 329)(88, 331)(96, 331)(97, 331)(102, 331)(103, 331)(109, 331)(112, 331)(114, 331)(115, 331)(116, 331)(120, 331)(121, 329)(121, 331)(134, 329)(134, 331)(140, 331)(155, 331)(169, 331)(170, 329)(170, 331)(186, 331)(187, 329)(187, 331)(189, 331)(193, 329)(193, 331)(199, 329)(199, 331)(209, 331)(226, 331)(234, 331)(237, 331)(242, 331)(251, 331)(261, 331)(284, 331)(301, 331)(319, 331)(322, 329)(322, 331)(323, 331)(324, 331)In [27]: #Clustering Coefficient clusterCoef = {} for k in edgeList: neighbours = edgeList[k] neighbLength = len(neighbours) denominator = (neighbLength\*(neighbLength-1))/2 numerator = 0for i in range(neighbLength): for j in range(i+1, neighbLength): if adjacencyMatrix[neighbours[i]][neighbours[j]] == 1: numerator += 1if denominator == 0.0: clusterCoef[k] = 0 else: clusterCoef[k] = numerator/denominator print('Clustering Coeffient of each nodes are : ',clusterCoef) print('\n') sumCC = 0**for** i **in** clusterCoef: sumCC += clusterCoef[i] print('Average Clustering Coefficent is ',sumCC/nodes)

Clustering Coeffient of each nodes are : {0: 1.0, 1: 1.0, 2: 1.0, 3: 0.6, 4: 1.0, 5: 1.0, 6: 1.0, 7: 0.1206896551724138, 8: 0.8333333333333334, 9: 0.833333333333334, 10: 1.0, 11: 1.0, 1 2: 0.12087912087912088, 13: 0, 14: 0, 15: 0.0, 16: 1.0, 17: 0, 18: 1.0, 19: 1.0, 20: 0, 21: 0.0, 22: 1.0, 23: 1.0, 24: 0, 25: 0.25, 26: 0, 27: 1.0, 28: 0.6666666666666666, 29: 1.0, 30: 6, 36: 1.0, 37: 0.66666666666666666, 38: 1.0, 39: 0.9047619047619048, 40: 1.0, 41: 0, 42: 0, 4 6666666, 48: 0.6666666666666666, 49: 0.777777777777778, 50: 1.0, 51: 0, 52: 0.8, 53: 0.90476 19047619048, 54: 0.8928571428571429, 55: 0.8, 56: 0.666666666666666, 57: 0.8214285714285714, 58: 0.64444444444445, 59: 0.6428571428571429, 60: 0.933333333333333, 61: 0.777777777777777 8, 62: 0.805555555555556, 63: 1.0, 64: 0.3329268292682927, 65: 0.6666666666666666, 66: 0.310 4722222222222, 71: 0.9047619047619048, 72: 0.666666666666666, 73: 0.8, 74: 0.5, 75: 0.533 333333333333, 76: 0.8333333333333334, 77: 1.0, 78: 1.0, 79: 0.7555555555555555, 80: 0.8, 81: 0.527272727272727, 82: 0.4380952380952381, 83: 1.0, 84: 0.5, 85: 1.0, 86: 0.66666666666666666 6, 87: 0, 88: 0, 89: 0.75, 90: 0.9673202614379085, 91: 0.9157894736842105, 92: 1.0, 93: 0.716 5775401069518, 94: 0.9090909090909091, 95: 0.8095238095238095, 96: 0.66666666666666666, 97: 1. 0, 98: 0.916666666666666, 99: 0.8095238095238095, 100: 0.8021978021978022, 101: 0.666666666 666666, 102: 1.0, 103: 0.6666666666666666, 104: 1.0, 105: 0.8666666666666667, 106: 1.0, 107: 0.833333333333334, 108: 0.6040816326530613, 109: 1.0, 110: 0.8095238095238095, 111: 0.400828 15734989646, 112: 1.0, 113: 0, 114: 1.0, 115: 0.6, 116: 0, 117: 0.14732561776665623, 118: 0.8 842364532019704, 119: 0.6785714285714286, 120: 0, 121: 0, 122: 0.6246153846153846, 123: 1.0, 124: 0.7714285714285715, 125: 0.8571428571428571, 126: 1.0, 127: 0.83333333333333334, 128: 1. 0, 129: 0.5555555555555556, 130: 0.67979797979798, 131: 1.0, 132: 0.8095238095238095, 133: 1.0, 134: 0, 135: 0.7, 136: 0.622222222222222, 137: 0.666666666666666, 138: 0.5357142857142 857, 139: 0.86666666666666667, 140: 1.0, 141: 0.36764705882352944, 142: 0.8205128205128205, 14 3: 0.2957334891876096, 144: 1.0, 145: 0.5444897959183673, 146: 0.4188150158299412, 147: 0.733 333333333333, 148: 1.0, 149: 0.5971014492753624, 150: 0.7, 151: 0.24296499656829101, 152: 0. 164021164021, 159: 1.0, 160: 0.5344827586206896, 161: 0.43892120571126386, 162: 0.33333333333 33333, 163: 1.0, 164: 0, 165: 0.2630252100840336, 166: 0.6670731707317074, 167: 0.85714285714 28571, 168: 0.9333333333333333, 169: 0.6666666666666666, 170: 0.0, 171: 0.7142857142857143, 1 72: 1.0, 173: 0.5275761973875182, 174: 0, 175: 0.46502732240437156, 176: 0.46365248226950356, 177: 0, 178: 0.6643437862950058, 179: 0, 180: 0.75, 181: 0.24593914436055822, 182: 0.73517786 56126482, 183: 1.0, 184: 0, 185: 0.8333333333333334, 186: 0, 187: 0, 188: 0.8496732026143791, 189: 1.0, 190: 1.0, 191: 0.6666666666666666, 192: 0, 193: 0, 194: 1.0, 195: 0, 196: 0.8157894 736842105, 197: 0.7575757575757576, 198: 1.0, 199: 0, 200: 0.2971905179982441, 201: 0.8011695 906432749, 202: 0.6923076923076923, 203: 0.7142857142857143, 204: 1.0, 205: 0.6, 206: 1.0, 20 7: 0, 208: 1.0, 209: 0, 210: 0, 211: 0.9590643274853801, 212: 0.6190476190476191, 213: 1.0, 2 14: 1.0, 215: 0.7, 216: 0.4915584415584416, 217: 0.7254901960784313, 218: 0.5518707482993197, 219: 1.0, 220: 0.4236734693877551, 221: 0.9333333333333333, 222: 1.0, 223: 0.8333333333333333 4, 224: 0.7802197802197802, 225: 1.0, 226: 0, 227: 1.0, 228: 0.5714285714285714, 229: 0.27105 052125100243, 230: 0, 231: 0.5359408033826638, 232: 0.6047430830039525, 233: 1.0, 234: 1.0, 2 35: 1.0, 236: 0.9, 237: 0, 238: 0.622222222222222, 239: 0.8928571428571429, 240: 0, 241: 0.7 142857142857143, 242: 1.0, 243: 0, 244: 0.8602941176470589, 245: 0.8023715415019763, 246: 0, 247: 0.4190531852717709, 248: 0.777777777777778, 249: 0.9120879120879121, 250: 0.80952380952 38095, 251: 1.0, 252: 0.8744588744588745, 253: 0.733333333333333, 254: 0.2401980198019802, 2 55: 0.6397058823529411, 256: 0, 257: 0.39887005649717516, 258: 0.83333333333333334, 259: 0.818 18181818182 260 0 18194987686513112 261 0 3974358974358974 262 0 6507936507936508 26

**Question 1** 

Convert text file to Adjacency Matrix and Edge List