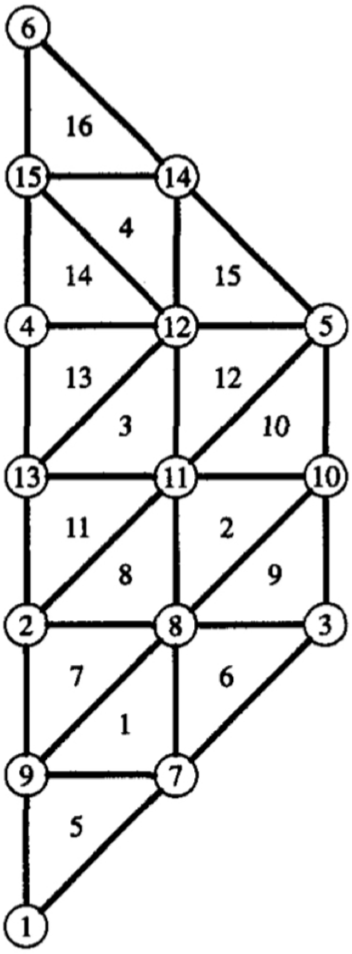
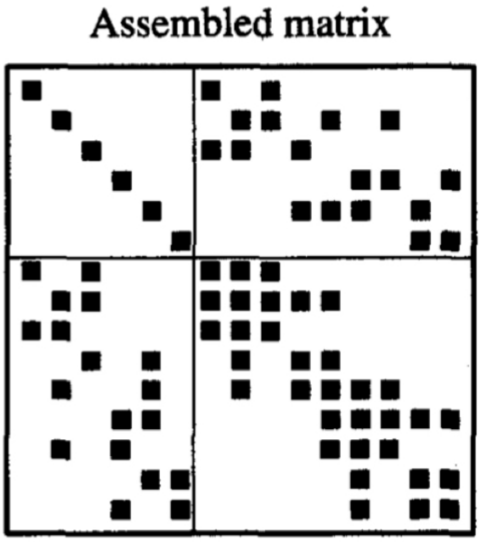
## Original adjacency matrix & graph

In [ ]:

```
graph = {
    1:[7,9],
    2:[8,9,11,13],
    3:[7,8,10],
    4:[12,13,15],
    5:[10,11,12,14],
    6:[14,15],
    7:[1,3,8,9],
    8:[2,3,7,9,10,11],
    9:[1,2,7,8],
    10:[3,5,8,11],
    11:[2,5,8,10,12,13],
    12:[4,5,11,13,14,15],
    13:[2,4,11,12],
    14:[5,6,12,15],
    15:[4,6,12,14]
}
```





102

## CMK Algorithm

In [ ]:

```
def CMK(graph, start):
    queue = []     # unvisited list
    queue.append(start)
    visited = set()     # visited list
    visited.add(start)
    new_queue = []
    while (len(queue) > 0):
        vertex = queue.pop(0)
        neighbors = graph[vertex]
        for w in neighbors:
            if w not in visited:
                queue.append(w)
                visited.add(w)
        new_queue.append(vertex)
    return new_queue
```

## Show new order and matrix

### Begin with 3

In [ ]:

```
M_CMK = CMK(graph, 3)
print("New order: ", M_CMK)
i = 1
Reorder = {}

# Reorder
for w in M_CMK:
    Reorder[M_CMK[i-1]] = i
    i = i + 1

# Create new matrix
Matrix = [[0 for _ in range(15)]
          for _ in range(15)]
for w in M_CMK:
    row = Reorder[w]-1
    Matrix[row][row] = 1
    neighbors = graph[w]
    for n in neighbors:
        column = Reorder[n]-1
        Matrix[row][column] = 1

for i in range(15):
    for j in range(15):
        if Matrix[i][j] == 1:
            print("■", end = " ")
        else:
            print("□", end = " ")
    print("\n")
```

```
New order:  [3, 7, 8, 10, 1, 9, 2, 11, 5, 13, 12, 14, 4, 15, 6]
■ ■ ■ ■ □ □ □ □ □ □ □ □ □ □ □

■ ■ ■ □ ■ ■ □ □ □ □ □ □ □ □ □

■ ■ ■ ■ □ ■ ■ ■ □ □ □ □ □ □ □

■ □ ■ ■ □ □ □ ■ ■ □ □ □ □ □ □

□ ■ □ □ ■ ■ □ □ □ □ □ □ □ □ □

□ ■ ■ □ ■ ■ ■ □ □ □ □ □ □ □ □

□ □ ■ □ □ ■ ■ ■ □ ■ □ □ □ □ □

□ □ ■ ■ □ □ ■ ■ ■ ■ ■ □ □ □ □

□ □ □ ■ □ □ □ ■ ■ □ ■ ■ □ □ □

□ □ □ □ □ □ ■ ■ □ ■ ■ □ ■ □ □

□ □ □ □ □ □ □ ■ ■ ■ ■ ■ ■ ■ □

□ □ □ □ □ □ □ □ ■ □ ■ ■ □ ■ ■

□ □ □ □ □ □ □ □ □ ■ ■ □ ■ ■ □

□ □ □ □ □ □ □ □ □ □ ■ ■ ■ ■ ■

□ □ □ □ □ □ □ □ □ □ □ ■ □ ■ ■
```

Bandwidth = 6

## Begin with 1

In [ ]:

```
M_CMK = CMK(graph, 1)
print("New order: ", M_CMK)
i = 1
Reorder = {}

# Reorder
for w in M_CMK:
    Reorder[M_CMK[i-1]] = i
    i = i + 1

# Create new matrix
Matrix = [[0 for _ in range(15)]
          for _ in range(15)]
for w in M_CMK:
    row = Reorder[w]-1
    Matrix[row][row] = 1
    neighbors = graph[w]
    for n in neighbors:
        column = Reorder[n]-1
        Matrix[row][column] = 1

for i in range(15):
    for j in range(15):
        if Matrix[i][j] == 1:
            print("■", end = " ")
        else:
            print("□", end = " ")
    print("\n")
```

```
New order:  [1, 7, 9, 3, 8, 2, 10, 11, 13, 5, 12, 4, 14, 15, 6]
■ ■ ■ □ □ □ □ □ □ □ □ □ □ □ □ 

■ ■ ■ ■ ■ □ □ □ □ □ □ □ □ □ □ 

■ ■ ■ □ ■ ■ □ □ □ □ □ □ □ □ □ 

□ ■ □ ■ ■ □ ■ □ □ □ □ □ □ □ □ 

□ ■ ■ ■ ■ ■ ■ ■ □ □ □ □ □ □ □ 

□ □ ■ □ ■ ■ □ ■ ■ □ □ □ □ □ □ 

□ □ □ ■ ■ □ ■ ■ □ ■ □ □ □ □ □ 

□ □ □ □ ■ ■ ■ ■ ■ ■ ■ □ □ □ □ 

□ □ □ □ □ ■ □ ■ ■ □ ■ ■ □ □ □ 

□ □ □ □ □ □ ■ ■ □ ■ ■ □ ■ □ □ 

□ □ □ □ □ □ □ ■ ■ ■ ■ ■ ■ ■ □ 

□ □ □ □ □ □ □ □ ■ □ ■ ■ □ ■ □ 

□ □ □ □ □ □ □ □ □ ■ ■ □ ■ ■ ■ 

□ □ □ □ □ □ □ □ □ □ ■ ■ ■ ■ ■ 

□ □ □ □ □ □ □ □ □ □ □ □ ■ ■ ■ 
```

Bandwidth = 4

# Reverse CMK(RCM)

### Begin with 1

```
In [ ]: M_CMK = CMK(graph,1)
        M_RCM = M_CMK[::-1]
        print(M_RCM)

        i = 1
        Reorder = {}

        # Reorder
        for w in M_RCM:
            Reorder[M_RCM[i-1]] = i
            i = i + 1

        # Create new matrix
        Matrix = [[0 for _ in range(15)]
                  for _ in range(15)]
        for w in M_RCM:
            row = Reorder[w]-1
            Matrix[row][row] = 1
            neighbors = graph[w]
            for n in neighbors:
                column = Reorder[n]-1
                Matrix[row][column] = 1

        for i in range(15):
            for j in range(15):
                if Matrix[i][j] == 1:
                    print("■", end = " ")
                else:
                    print("□", end = " ")
            print("\n")
```

```
[6, 15, 14, 4, 12, 5, 13, 11, 10, 2, 8, 3, 9, 7, 1]
■ ■ ■ □ □ □ □ □ □ □ □ □ □ □ □ 

■ ■ ■ ■ ■ □ □ □ □ □ □ □ □ □ □ 

■ ■ ■ □ ■ ■ □ □ □ □ □ □ □ □ □ 

□ ■ □ ■ ■ □ ■ □ □ □ □ □ □ □ □ 

□ ■ ■ ■ ■ ■ ■ ■ □ □ □ □ □ □ □ 

□ □ ■ □ ■ ■ □ ■ ■ □ □ □ □ □ □ 

□ □ □ ■ ■ □ ■ ■ □ ■ □ □ □ □ □ 

□ □ □ □ ■ ■ ■ ■ ■ ■ ■ □ □ □ □ 

□ □ □ □ □ ■ □ ■ ■ □ ■ ■ □ □ □ 

□ □ □ □ □ □ ■ ■ □ ■ ■ □ ■ □ □ 

□ □ □ □ □ □ □ ■ ■ ■ ■ ■ ■ ■ □ 

□ □ □ □ □ □ □ □ ■ □ ■ ■ □ ■ □ 

□ □ □ □ □ □ □ □ □ ■ ■ □ ■ ■ ■ 

□ □ □ □ □ □ □ □ □ □ ■ ■ ■ ■ ■ 

□ □ □ □ □ □ □ □ □ □ □ □ ■ ■ ■ 
```

Bandwidth = 4

## Test with MEDO single pin cell

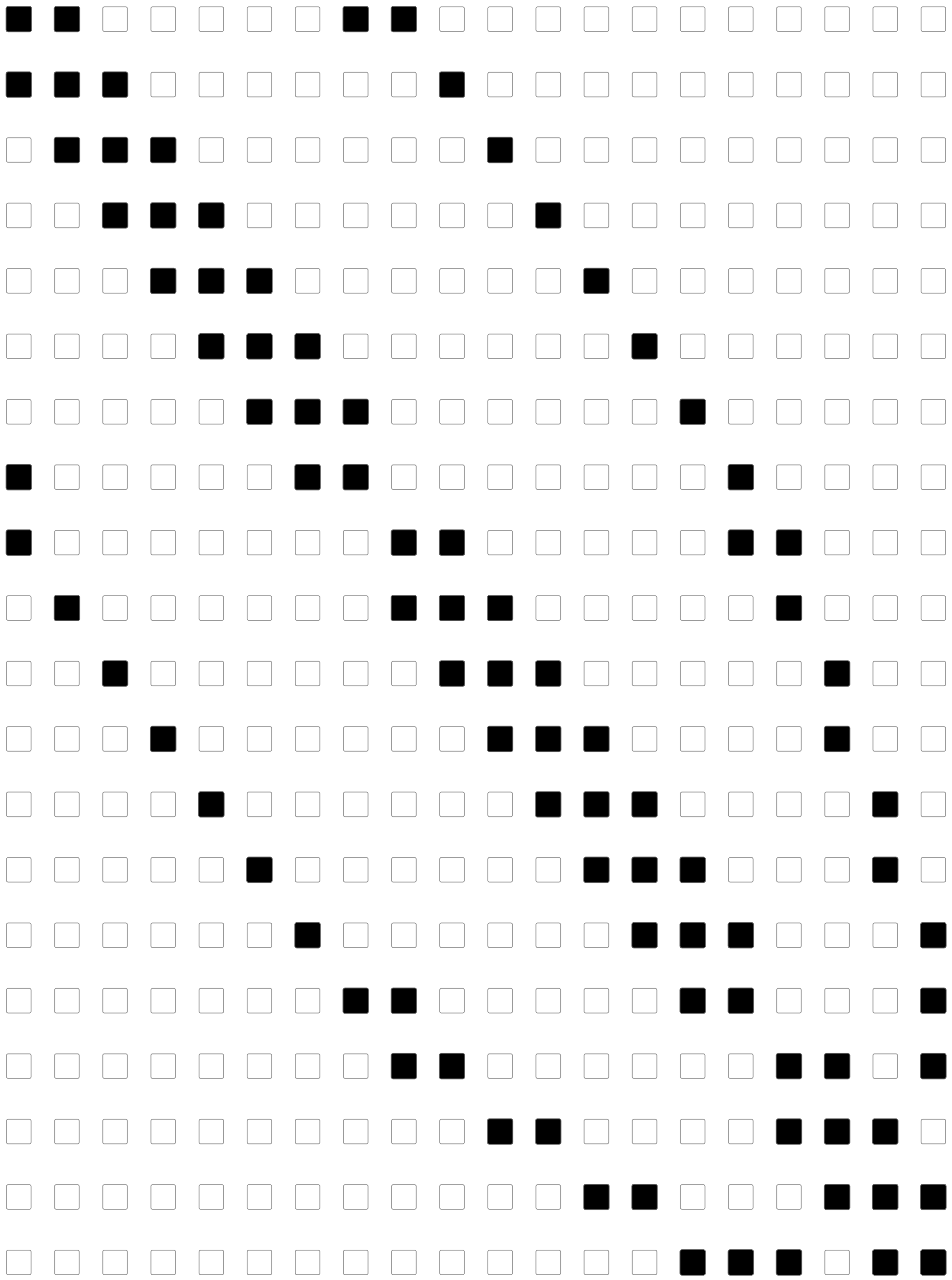
In [ ]:

```
graph = {
    0:[1,7,8],
    1:[0,2,9],
    2:[1,3,10],
    3:[2,4,11],
    4:[3,5,12],
    5:[4,6,13],
    6:[5,7,14],
    7:[0,6,15],
    8:[0,9,15,16],
    9:[1,8,10,16],
    10:[2,9,11,17],
    11:[3,10,12,17],
    12:[4,11,13,18],
    13:[5,12,14,18],
    14:[6,13,15,19],
    15:[7,8,14,19],
    16:[8,9,17,19],
    17:[10,11,16,18],
    18:[12,13,17,19],
    19:[14,15,16,18]
}
```

## Show the origin matrix shape

In [ ]:

```
Matrix = [[0 for _ in range(20)]
          for _ in range(20)]
for w in graph:
    Matrix[w][w] = 1
    neighbors = graph[w]
    for n in neighbors:
        Matrix[w][n] = 1
for i in range(20):
    for j in range(20):
        if Matrix[i][j] == 1:
            print("■", end = " ")
        else:
            print("□", end = " ")
    print("\n")
```



## CMK

In [ ]:

```
# Begin with 0
M_CMK = CMK(graph, 0)
print("New order: ", M_CMK)

i = 0
Reorder = {}

# Reorder
for w in M_CMK:
    Reorder[M_CMK[i]] = i
    i = i + 1

# Create new matrix
Matrix = [[0 for _ in range(20)]
          for _ in range(20)]
for w in M_CMK:
    row = Reorder[w]
    Matrix[row][row] = 1
    neighbors = graph[w]
    for n in neighbors:
        column = Reorder[n]
        Matrix[row][column] = 1

for i in range(20):
    for j in range(20):
        if Matrix[i][j] == 1:
            print("■", end = " ")
        else:
            print("□", end = " ")
    print("\n")
```

```
New order:  [0, 1, 7, 8, 2, 9, 6, 15, 16, 3, 10, 5, 14, 19, 17, 4, 11, 13, 18, 1
2]
■ ■ ■ ■ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

■ ■ □ □ ■ ■ □ □ □ □ □ □ □ □ □ □ □ □ □ □

■ □ ■ □ □ □ ■ ■ □ □ □ □ □ □ □ □ □ □ □ □

■ □ □ ■ □ ■ □ ■ ■ □ □ □ □ □ □ □ □ □ □ □

□ ■ □ □ ■ □ □ □ □ ■ ■ □ □ □ □ □ □ □ □ □

□ ■ □ ■ □ ■ □ □ ■ □ ■ □ □ □ □ □ □ □ □ □

□ □ ■ □ □ □ ■ □ □ □ □ ■ ■ □ □ □ □ □ □ □

□ □ ■ ■ □ □ □ ■ □ □ □ □ ■ ■ □ □ □ □ □ □

□ □ □ ■ □ ■ □ □ ■ □ □ □ □ ■ ■ □ □ □ □ □

□ □ □ □ ■ □ □ □ □ ■ □ □ □ □ □ ■ ■ □ □ □

□ □ □ □ ■ ■ □ □ □ □ ■ □ □ □ ■ □ ■ □ □ □

□ □ □ □ □ □ ■ □ □ □ □ ■ □ □ □ ■ □ ■ □ □

□ □ □ □ □ □ ■ ■ □ □ □ □ ■ ■ □ □ □ ■ □ □

□ □ □ □ □ □ □ ■ ■ □ □ □ ■ ■ □ □ □ □ ■ □

□ □ □ □ □ □ □ □ ■ □ ■ □ □ □ ■ □ ■ □ ■ □

□ □ □ □ □ □ □ □ □ ■ □ ■ □ □ □ ■ □ □ □ ■

□ □ □ □ □ □ □ □ □ ■ ■ □ □ □ ■ □ ■ □ □ ■

□ □ □ □ □ □ □ □ □ □ □ ■ ■ □ □ □ □ ■ ■ ■

□ □ □ □ □ □ □ □ □ □ □ □ □ ■ ■ □ □ ■ ■ ■

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ ■ ■ ■ ■ ■
```

## RCM

In [ ]:

```
# Begin with 0
M_CMK = CMK(graph,1)
M_RCM = M_CMK[::-1]
print(M_RCM)

i = 0
Reorder = {}

# Reorder
for w in M_RCM:
    Reorder[M_RCM[i]] = i
    i = i + 1

# Create new matrix
Matrix = [[0 for _ in range(20)]
          for _ in range(20)]
for w in M_RCM:
    row = Reorder[w]
    Matrix[row][row] = 1
    neighbors = graph[w]
    for n in neighbors:
        column = Reorder[n]
        Matrix[row][column] = 1

for i in range(20):
    for j in range(20):
        if Matrix[i][j] == 1:
            print("■", end = " ")
        else:
            print("□", end = " ")
    print("\n")
```

```
[13, 18, 12, 14, 5, 19, 17, 11, 4, 15, 6, 16, 10, 3, 8, 7, 9, 2, 0, 1]
■ ■ ■ ■ ■ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
■ ■ ■ □ □ ■ ■ □ □ □ □ □ □ □ □ □ □ □ □ □
■ ■ ■ □ □ □ □ ■ ■ □ □ □ □ □ □ □ □ □ □ □
■ □ □ ■ □ ■ □ □ □ ■ ■ □ □ □ □ □ □ □ □ □
■ □ □ □ ■ □ □ □ ■ □ ■ □ □ □ □ □ □ □ □ □
□ ■ □ ■ □ ■ □ □ □ ■ □ ■ □ □ □ □ □ □ □ □
□ ■ □ □ □ □ ■ ■ □ □ □ ■ ■ □ □ □ □ □ □ □
□ □ ■ □ □ □ ■ ■ □ □ □ □ ■ ■ □ □ □ □ □ □
□ □ ■ □ ■ □ □ □ ■ □ □ □ □ ■ □ □ □ □ □ □
□ □ □ ■ □ ■ □ □ □ ■ □ □ □ □ ■ ■ □ □ □ □
□ □ □ ■ ■ □ □ □ □ □ ■ □ □ □ □ ■ □ □ □ □
□ □ □ □ □ ■ ■ □ □ □ □ ■ □ □ ■ □ ■ □ □ □
□ □ □ □ □ □ ■ ■ □ □ □ □ ■ □ □ □ ■ ■ □ □
□ □ □ □ □ □ □ ■ ■ □ □ □ □ ■ □ □ □ ■ □ □
□ □ □ □ □ □ □ □ □ ■ □ ■ □ □ ■ □ ■ □ ■ □
□ □ □ □ □ □ □ □ □ ■ ■ □ □ □ □ ■ □ □ ■ □
□ □ □ □ □ □ □ □ □ □ □ ■ ■ □ ■ □ ■ □ □ ■
□ □ □ □ □ □ □ □ □ □ □ □ ■ ■ □ □ □ ■ □ ■
□ □ □ □ □ □ □ □ □ □ □ □ □ □ ■ ■ □ □ ■ ■
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ ■ ■ ■ ■
```

In [ ]: