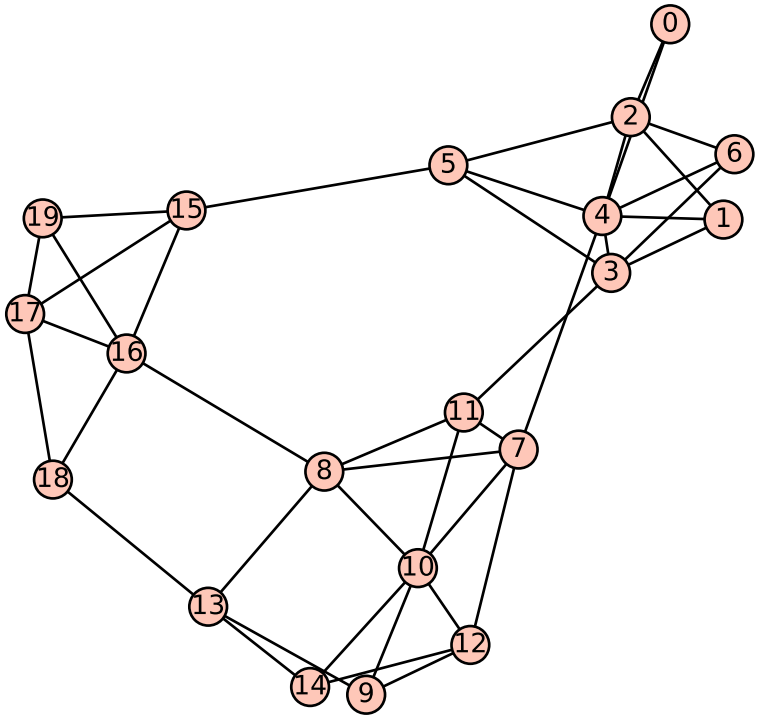
# 2020-08-04-143949

Ido Doron

8/6/2020

```
M = [[0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[
0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[
1,1,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0],[
0,1,0,0,1,1,1,0,0,0,0,1,0,0,0,0,0,0,0,0],[
1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0],[
0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0],[
0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[
0,0,0,0,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0],[
0,0,0,0,0,0,0,1,0,0,1,1,0,1,0,0,1,0,0,0],[
0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0],[
0,0,0,0,0,0,0,0,1,1,1,0,1,1,0,1,0,0,0,0],[
0,0,0,1,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,0],[
0,0,0,0,0,0,0,1,0,1,1,0,0,0,1,0,0,0,0,0],[
0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,1,0],[
0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0],[
0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,1],[
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,1,1],[
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1],[
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,1,0,0],[
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0]]

E = []
for i in range(20):
    for j in range(20):
        if i <= j and M[i][j] == 1:
            E.append((i,j))

G = Graph(E)
P = G.plot()
P.show()
s = "{"
delimiter = ""
for (i,j) in G.edges(False, True):
    s += delimiter + "{" + str(i) + "," + str(j) + "}"
    delimiter = ","
```

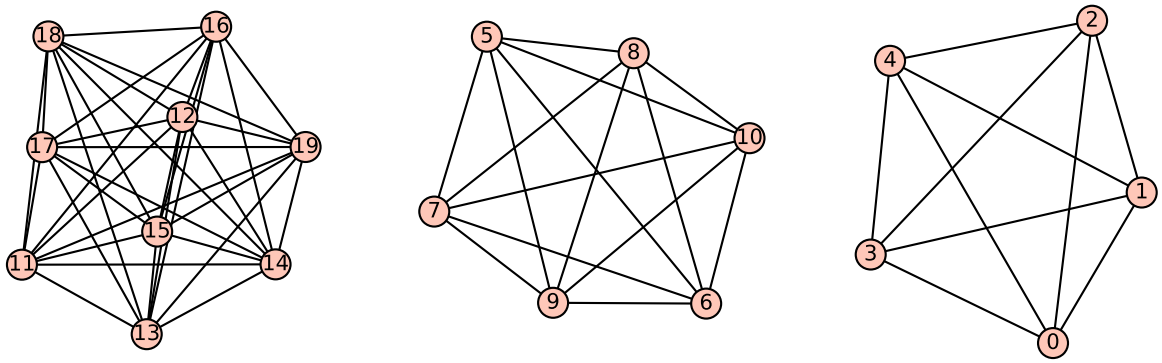
```
s += "}"
print(str(s))
len("number of edges: " + str(G.edges(False, True)))

adjmat = str(G.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print("\n20\n20\n" + adjmat)
```




{{0,2},{0,4},{1,2},{1,3},{1,4},{2,4},{2,5},{2,6},{3,4},{3,5},{3,6},{3,11},{4,5},{4,6},{4,7
},{5,15},{7,8},{7,10},{7,11},{7,12},{8,10},{8,11},{8,13},{8,16},{9,10},{9,12},{9,13},{10,1
1},{10,12},{10,14},{12,14},{13,14},{13,18},{15,16},{15,17},{15,19},{16,17},{16,18},{16,19}
,{17,18},{17,19}}
385
20
20
0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 0

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

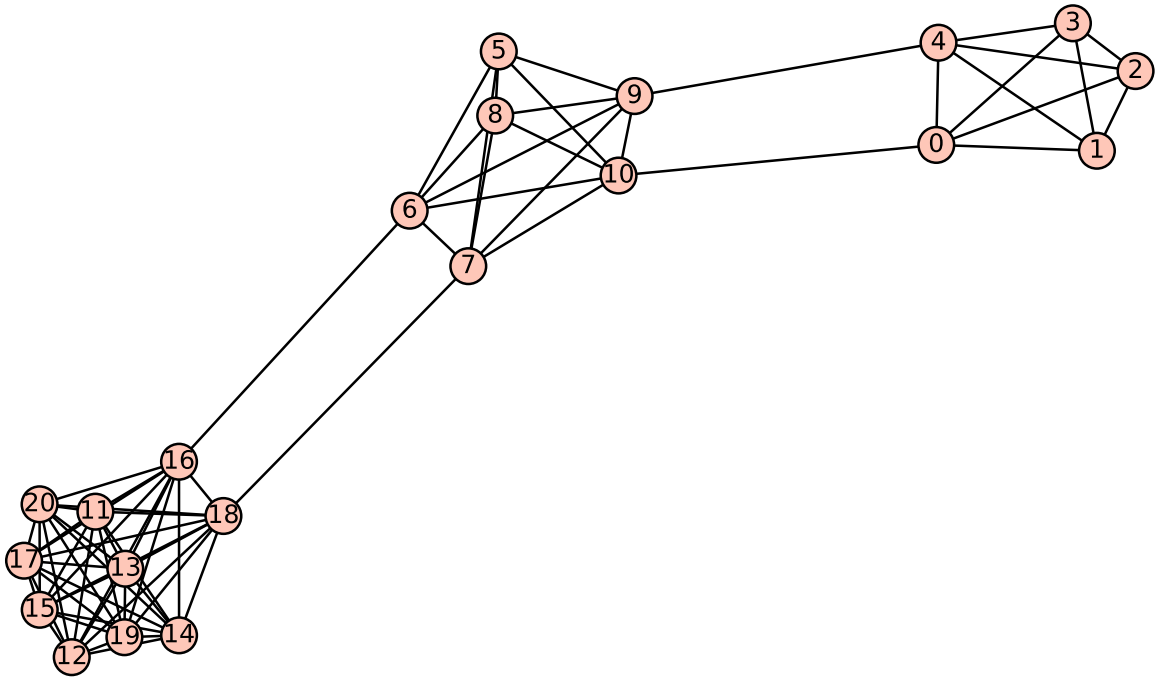
```
G1 = graphs.CompleteGraph(5)
G2 = graphs.CompleteGraph(6)
G3 = graphs.CompleteGraph(9)
G1 = G1.disjoint_union(G2, labels='integers')
G1 = G1.disjoint_union(G3, labels='integers')
P= G1.plot()
P.show()
s = "{"
delimiter = ""
for (i,j) in G1.edges(False, True):
    s += delimiter + "{" + str(i) + "," + str(j) + "}"
    delimiter = ","
s += "}"
print(str(s))
len("number of edges: " + str(G1.edges(False, True)))

adjmat = str(G1.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print("\n20\n20\n" + adjmat)
```




{{0,1},{0,2},{0,3},{0,4},{1,2},{1,3},{1,4},{2,3},{2,4},{3,4},{5,6},{5,7},{5,8},{5,9},{5,10},{6,7},{6,8},{6,9},{6,10},{7,8},{7,9},{7,10},{8,9},{8,10},{9,10},{11,12},{11,13},{11,14},{11,15},{11,16},{11,17},{11,18},{11,19},{12,13},{12,14},{12,15},{12,16},{12,17},{12,18},{12,19},{13,14},{13,15},{13,16},{13,17},{13,18},{13,19},{14,15},{14,16},{14,17},{14,18},{14,19},{15,16},{15,17},{15,18},{15,19},{16,17},{16,18},{16,19},{17,18},{17,19},{18,19}}
582

20
20
0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0

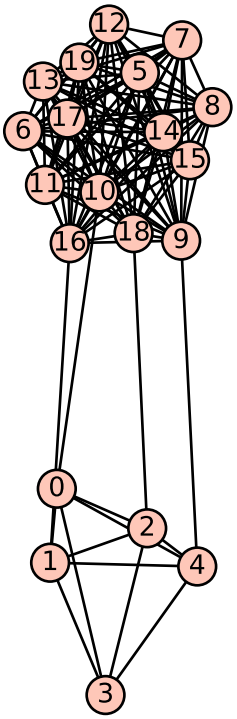
```
G1 = graphs.CompleteGraph(5)
G2 = graphs.CompleteGraph(6)
G3 = graphs.CompleteGraph(10)
G1 = G1.disjoint_union(G2, labels='integers')
G1 = G1.disjoint_union(G3, labels='integers')
G1.add_edges([(6,16),(7,18), (4,9), (0,10)])
P= G1.plot()
P.show()
s = "{"
delimiter = ""
for (i,j) in G1.edges(False, True):
    s += delimiter + "{" + str(i) + "," + str(j) + "}"
    delimiter = ","
s += "}"
print(str(s))
len("number of edges: " + str(G1.edges(False, True)))

adjmat = str(G1.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print("\n20\n20\n" + adjmat)
```




{{0,1},{0,2},{0,3},{0,4},{0,10},{1,2},{1,3},{1,4},{2,3},{2,4},{3,4},{4,9},{5,6},{5,7},{5,8
},{5,9},{5,10},{6,7},{6,8},{6,9},{6,10},{6,16},{7,8},{7,9},{7,10},{7,18},{8,9},{8,10},{9,1
0},{11,12},{11,13},{11,14},{11,15},{11,16},{11,17},{11,18},{11,19},{11,20},{12,13},{12,14}
,{12,15},{12,16},{12,17},{12,18},{12,19},{12,20},{13,14},{13,15},{13,16},{13,17},{13,18},{
13,19},{13,20},{14,15},{14,16},{14,17},{14,18},{14,19},{14,20},{15,16},{15,17},{15,18},{15
,19},{15,20},{16,17},{16,18},{16,19},{16,20},{17,18},{17,19},{17,20},{18,19},{18,20},{19,2
0}}
707
20
20
0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1
0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 0 1 1 1 1

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

```
G1 = graphs.CompleteGraph(5)
G2 = graphs.CompleteGraph(15)
G1 = G1.disjoint_union(G2, labels='integers')
G1.add_edges([(1,16),(2,18), (4,9), (0,10)])
P= G1.plot()
P.show()
s = "{"
delimiter = ""
for (i,j) in G1.edges(False, True):
    s += delimiter + "{" + str(i) + "," + str(j) + "}"
    delimiter = ","
s += "}"
print(str(s))
len("number of edges: " + str(G1.edges(False, True)))

adjmat = str(G1.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print("\n20\n20\n" + adjmat)
```

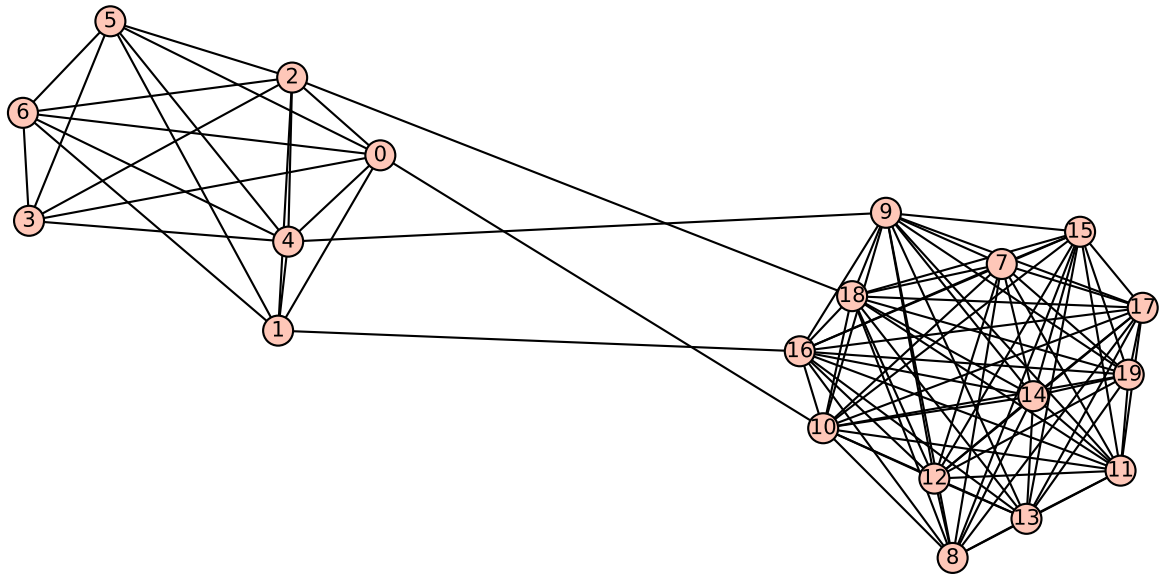



{{0,1},{0,2},{0,3},{0,4},{0,10},{1,2},{1,3},{1,4},{1,16},{2,3},{2,4},{2,18},{3,4},{4,9},{5
,6},{5,7},{5,8},{5,9},{5,10},{5,11},{5,12},{5,13},{5,14},{5,15},{5,16},{5,17},{5,18},{5,19

},{6,7},{6,8},{6,9},{6,10},{6,11},{6,12},{6,13},{6,14},{6,15},{6,16},{6,17},{6,18},{6,19},
{7,8},{7,9},{7,10},{7,11},{7,12},{7,13},{7,14},{7,15},{7,16},{7,17},{7,18},{7,19},{8,9},{8
,10},{8,11},{8,12},{8,13},{8,14},{8,15},{8,16},{8,17},{8,18},{8,19},{9,10},{9,11},{9,12},{
9,13},{9,14},{9,15},{9,16},{9,17},{9,18},{9,19},{10,11},{10,12},{10,13},{10,14},{10,15},{1
0,16},{10,17},{10,18},{10,19},{11,12},{11,13},{11,14},{11,15},{11,16},{11,17},{11,18},{11,
19},{12,13},{12,14},{12,15},{12,16},{12,17},{12,18},{12,19},{13,14},{13,15},{13,16},{13,17
},{13,18},{13,19},{14,15},{14,16},{14,17},{14,18},{14,19},{15,16},{15,17},{15,18},{15,19},
{16,17},{16,18},{16,19},{17,18},{17,19},{18,19}}
1112
20
20
0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0

```
G1 = graphs.CompleteGraph(7)
G2 = graphs.CompleteGraph(13)
G1 = G1.disjoint_union(G2, labels='integers')
G1.add_edges([(1,16),(2,18), (4,9), (0,10)])
G1.delete_edges([(8,19), (7,13), (1,3)])
P= G1.plot()
P.show()
s = "{"
delimiter = ""
for (i,j) in G1.edges(False, True):
    s += delimiter + "{" + str(i) + "," + str(j) + "}"
    delimiter = ","
s += "}"
print(str(s))
len("number of edges: " + str(G1.edges(False, True)))
```

```
adjmat = str(G1.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print("\n20\n20\n" + adjmat)
```




{{0,1},{0,2},{0,3},{0,4},{0,5},{0,6},{0,10},{1,2},{1,4},{1,5},{1,6},{1,16},{2,3},{2,4},{2,5},{2,6},{2,18},{3,4},{3,5},{3,6},{4,5},{4,6},{4,9},{5,6},{7,8},{7,9},{7,10},{7,11},{7,12},{7,14},{7,15},{7,16},{7,17},{7,18},{7,19},{8,9},{8,10},{8,11},{8,12},{8,13},{8,14},{8,15},{8,16},{8,17},{8,18},{9,10},{9,11},{9,12},{9,13},{9,14},{9,15},{9,16},{9,17},{9,18},{9,19},{10,11},{10,12},{10,13},{10,14},{10,15},{10,16},{10,17},{10,18},{10,19},{11,12},{11,13},{11,14},{11,15},{11,16},{11,17},{11,18},{11,19},{12,13},{12,14},{12,15},{12,16},{12,17},{12,18},{12,19},{13,14},{13,15},{13,16},{13,17},{13,18},{13,19},{14,15},{14,16},{14,17},{14,18},{14,19},{15,16},{15,17},{15,18},{15,19},{16,17},{16,18},{16,19},{17,18},{17,19},{18,19}}}
938
20
20
0 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0
1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0
1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0
1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1

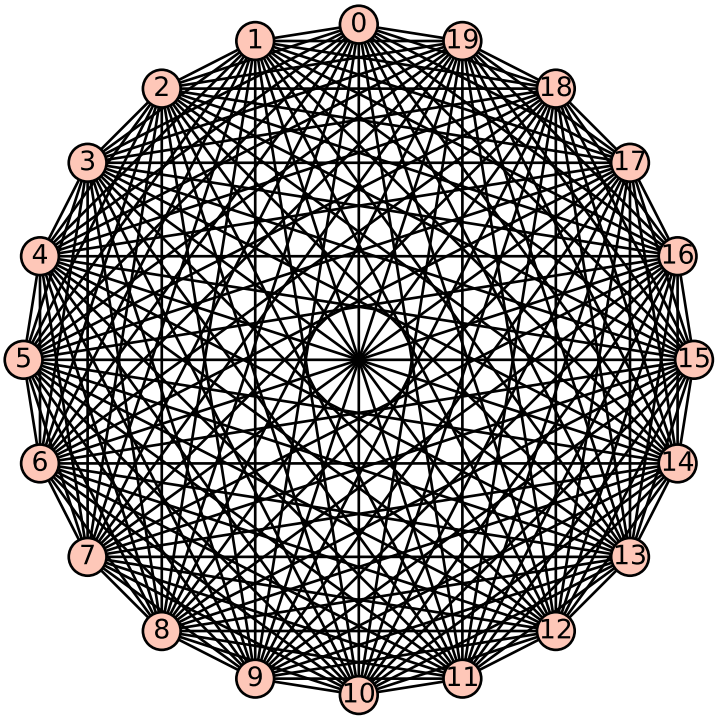
0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1
0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1
0 0 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1
0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0

```
G1 = Graph()
for i in range(20):
    G1.add_vertices([i])
P= G1.plot()
P.show()
s = "{"
delimiter = ""
for (i,j) in G1.edges(False, True):
    s += delimiter + "{" + str(i) + "," + str(j) + "}"
    delimiter = ","
s += "}"
print(str(s))
len("number of edges: " + str(G1.edges(False, True)))

adjmat = str(G1.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print("\n20\n20\n" + adjmat)
```


{}
19
20
20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

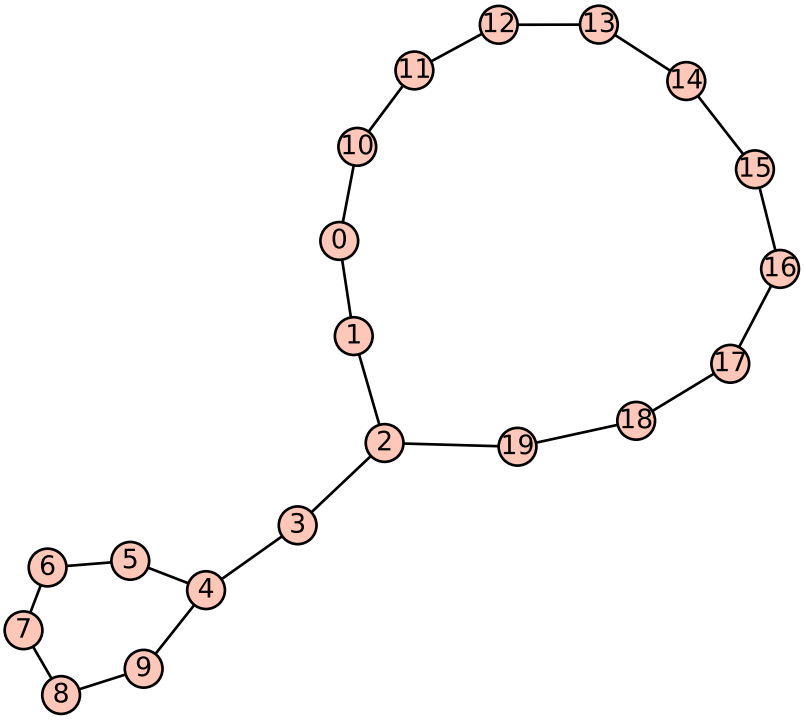
```
G1 = graphs.CompleteGraph(20)
P= G1.plot()
P.show()
s = ”{”
delimiter = ””
for (i,j) in G1.edges(False, True):
    s += delimiter + ”{” + str(i) + ”,” + str(j) + ”}”
    delimiter = ”,”
s += ”}”
print(str(s))
len(”number of edges: ” + str(G1.edges(False, True)))

adjmat = str(G1.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print(”\n20\n20\n” + adjmat)
```




{{0,1},{0,2},{0,3},{0,4},{0,5},{0,6},{0,7},{0,8},{0,9},{0,10},{0,11},{0,12},{0,13},{0,14},
{0,15},{0,16},{0,17},{0,18},{0,19},{1,2},{1,3},{1,4},{1,5},{1,6},{1,7},{1,8},{1,9},{1,10},
{1,11},{1,12},{1,13},{1,14},{1,15},{1,16},{1,17},{1,18},{1,19},{2,3},{2,4},{2,5},{2,6},{2,
7},{2,8},{2,9},{2,10},{2,11},{2,12},{2,13},{2,14},{2,15},{2,16},{2,17},{2,18},{2,19},{3,4}
,{3,5},{3,6},{3,7},{3,8},{3,9},{3,10},{3,11},{3,12},{3,13},{3,14},{3,15},{3,16},{3,17},{3,
18},{3,19},{4,5},{4,6},{4,7},{4,8},{4,9},{4,10},{4,11},{4,12},{4,13},{4,14},{4,15},{4,16},
{4,17},{4,18},{4,19},{5,6},{5,7},{5,8},{5,9},{5,10},{5,11},{5,12},{5,13},{5,14},{5,15},{5,
16},{5,17},{5,18},{5,19},{6,7},{6,8},{6,9},{6,10},{6,11},{6,12},{6,13},{6,14},{6,15},{6,16
},{6,17},{6,18},{6,19},{7,8},{7,9},{7,10},{7,11},{7,12},{7,13},{7,14},{7,15},{7,16},{7,17}
,{7,18},{7,19},{8,9},{8,10},{8,11},{8,12},{8,13},{8,14},{8,15},{8,16},{8,17},{8,18},{8,19}
,{9,10},{9,11},{9,12},{9,13},{9,14},{9,15},{9,16},{9,17},{9,18},{9,19},{10,11},{10,12},{10
,13},{10,14},{10,15},{10,16},{10,17},{10,18},{10,19},{11,12},{11,13},{11,14},{11,15},{11,1
6},{11,17},{11,18},{11,19},{12,13},{12,14},{12,15},{12,16},{12,17},{12,18},{12,19},{13,14}
,{13,15},{13,16},{13,17},{13,18},{13,19},{14,15},{14,16},{14,17},{14,18},{14,19},{15,16},{
15,17},{15,18},{15,19},{16,17},{16,18},{16,19},{17,18},{17,19},{18,19}}
1727
20
20
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1

```
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
```

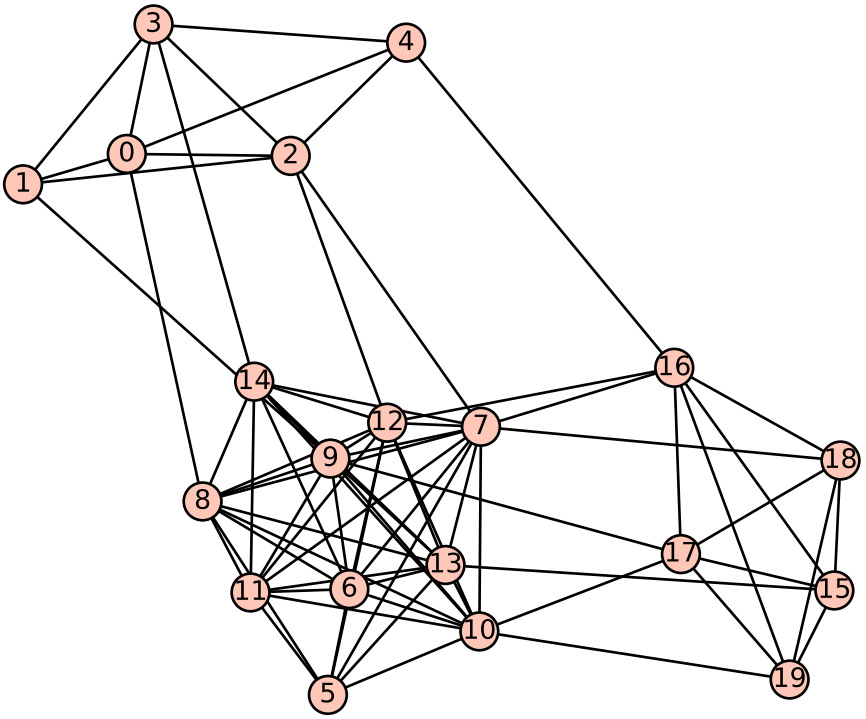
```
G1 = graphs.PathGraph(10)
G2 = graphs.PathGraph(10)
G1 = G1.disjoint_union(G2, labels='integers')
G1.add_edges([(2,19), (4,9), (0,10)])
P= G1.plot()
P.show()
s = "{"
delimiter = ""
for (i,j) in G1.edges(False, True):
    s += delimiter + "{" + str(i) + "," + str(j) + "}"
    delimiter = ","
s += "}"
print(str(s))
len("number of edges: " + str(G1.edges(False, True)))

adjmat = str(G1.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print("\n20\n20\n" + adjmat)
```




{{0,1},{0,10},{1,2},{2,3},{2,19},{3,4},{4,5},{4,9},{5,6},{6,7},{7,8},{8,9},{10,11},{11,12}
,{12,13},{13,14},{14,15},{15,16},{16,17},{17,18},{18,19}}
205
20
20
0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

```
M = [[0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,0.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 \
    ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [1.0 ,0.0 ,1.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 \
    ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [1.0 ,1.0 ,0.0 ,1.0 ,1.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 \
    ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [1.0 ,1.0 ,1.0 ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 \
    ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [1.0 ,0.0 ,1.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 \
    ,0.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,0.0 ,1.0 ,1.0 ,1.0 \
    ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 \
    ,1.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,1.0 ,1.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 \
    ,1.0 ,1.0 ,0.0 ,1.0 ,0.0 ,1.0 ,0.0 ],
 [1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 \
    ,1.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [0.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,0.0 ,1.0 ,1.0 ,1.0 \
    ,1.0 ,1.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,0.0 ,1.0 ,1.0 \
    ,1.0 ,1.0 ,0.0 ,0.0 ,1.0 ,0.0 ,1.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,0.0 ,1.0 \
    ,1.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,0.0 \
    ,1.0 ,1.0 ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 \
    ,0.0 ,1.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [0.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,1.0 \
    ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 \
    ,1.0 ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 \
    ,0.0 ,0.0 ,1.0 ,0.0 ,1.0 ,1.0 ,1.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,1.0 ,0.0 ,0.0 \
    ,0.0 ,0.0 ,1.0 ,1.0 ,0.0 ,1.0 ,1.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 \
    ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,0.0 ,1.0 ],
 [0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,0.0 ,1.0 ,0.0 ,0.0 \
    ,0.0 ,0.0 ,1.0 ,1.0 ,1.0 ,1.0 ,0.0 ]]

E = []
for i in range(20):
    for j in range(20):
        if i < j and M[i][j] == 1:
            E.append((i,j))
```

```
G = Graph(E)
P = G.plot()
P.show()
s = "{"
delimiter = ""
for (i,j) in G.edges(False, True):
    s += delimiter + "{" + str(i) + "," + str(j) + "}"
    delimiter = ","
s += "}"
print(str(s))
len("number of edges: " + str(G.edges(False, True)))

adjmat = str(G.adjacency_matrix())
adjmat = adjmat.replace('[', '')
adjmat = adjmat.replace(']', '')
print("\n20\n20\n" + adjmat)
```




{{0,1},{0,2},{0,3},{0,4},{0,8},{1,2},{1,3},{1,9},{2,3},{2,4},{2,7},{2,12},{3,4},{3,14},{4,16},{5,6},{5,7},{5,8},{5,10},{5,11},{5,12},{5,13},{6,7},{6,8},{6,9},{6,10},{6,11},{6,12},{6,13},{6,14},{7,8},{7,9},{7,10},{7,11},{7,12},{7,13},{7,14},{7,16},{7,18},{8,9},{8,10},{8,11},{8,12},{8,13},{8,14},{9,10},{9,11},{9,12},{9,13},{9,14},{9,17},{10,11},{10,12},{10,13},{10,14},{10,17},{10,19},{11,12},{11,13},{11,14},{12,13},{12,14},{12,16},{13,14},{13,15},{15,16},{15,17},{15,18},{15,19},{16,17},{16,18},{16,19},{17,18},{17,19},{18,19}}
695
20
20
0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

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