

# 2016-01-25-151740

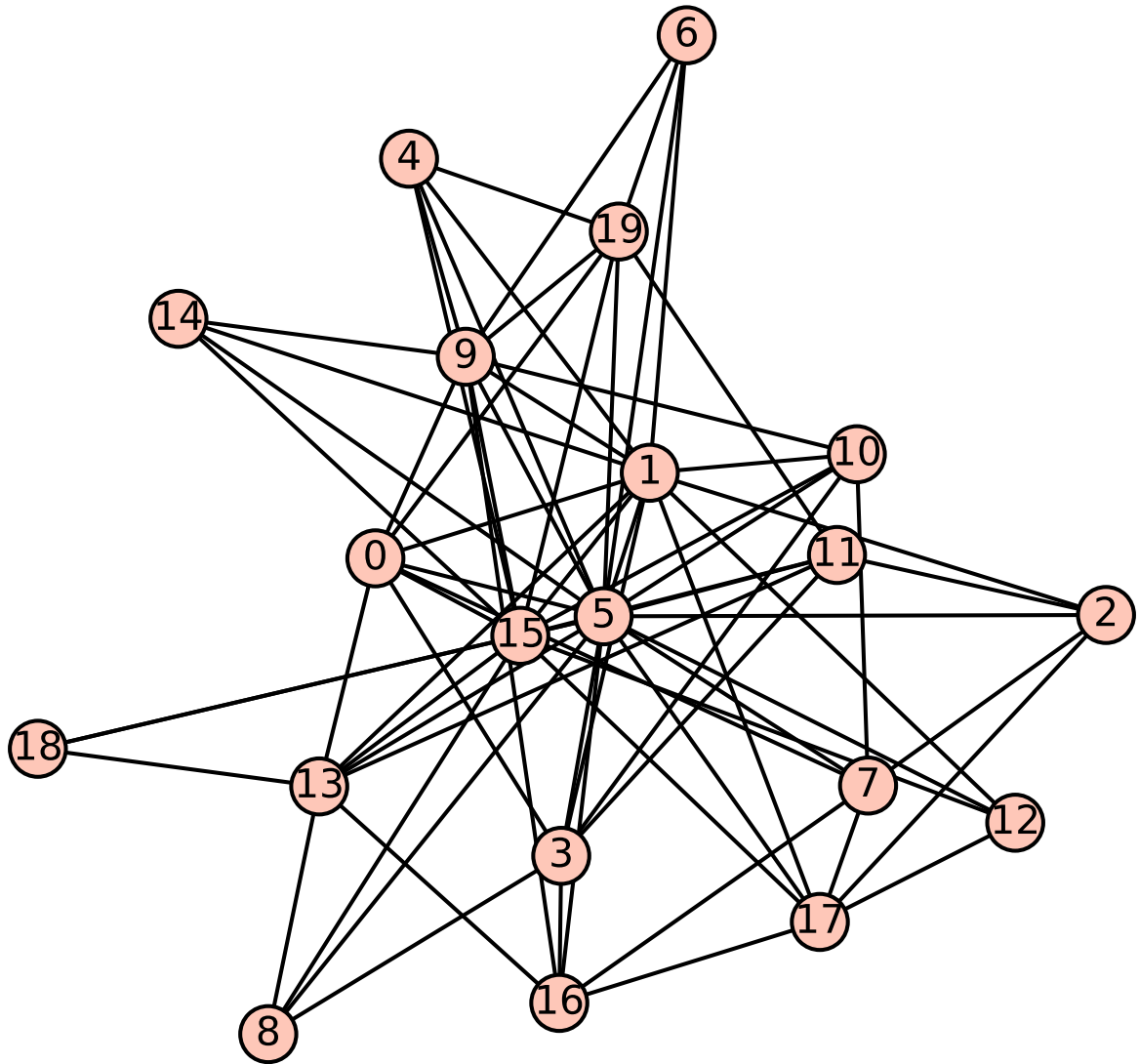
Matthew Conroy

1/27/2016

```
var('n')
n=20 # we will generate a graph with n vertices
A=matrix(n) # A is the nxn adjacency matrix for the graph
# define A
for i in range(2,n+2): # so i=2,3, ..., n+1
    for j in range(2,n+2): # so j=2,3, ..., n+1
        # if (i,j) satisfies some crazy arithmetic relationship...
        if ((i!=j) and is_prime(i^2+j^2+i*j+4)):
            # then set the corresponding entry in A to 1
            A[i-2,j-2]=1
print A.str() # print A
B=Graph(A) # define B to be the graph whose adjacency matrix is A
# Thus, B is the graph on {2,3,...,n} with an edge (u,v) iff u^2+v^2+i*j+4 is prime
B.plot() # why not draw a picture of B?
# note that a vertex labeled m in the plot actually corresponds to the integer m+2
# so 0 is 2, 1 is 3, etc.
B.chromatic_number() # have Sage find the chromatic number of B
n
[0 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 0 0 1]
[1 0 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 0 0]
[0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0]
[1 1 0 0 0 1 0 0 1 0 1 1 0 0 0 0 1 0 0 0]
[0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0]
[1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
[1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0]
[0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0]
[1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1]
[0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0]
[0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1]
[0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0]
[1 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 0 1 0]
```

---

```
[0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0]
[1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1]
[0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0]
[0 1 1 0 0 1 0 1 0 0 0 0 1 0 0 1 1 0 0 0]
[0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0]
[1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0 0 0]
```



5