PositiveNegativeSquareEnergiesGraphs

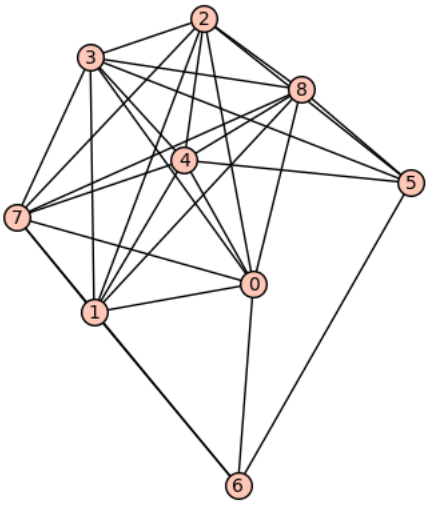
last edited Mar 14, 2023, 9:45:00 AM by hogben


```
###################################
# This Sage worksheet is written by Aida Abiad, Leonardo de Lima, Krystal Guo, and Leslie Hogben
hogben@aimath.org
# This worksheet contains computations for the paper
# Positive and Negative Square Energies of Graphs
# by
# A. Abiad, L. de Lima, D.N. Desai, K. Guo, L. Hogben, and J. Madrid
#
# The order of the cells is
#    Function Cell (must be run first)
#    Graph for Figure 3.2
#    Computation for Propostion 3.9
#    Data for Table 4.1
#    Data fpr Table 4.2
# Reminder: To run computations, you must first enter the Functions Cell (the first cell after this one)
# Then within each topic you must enter the cells in order.
```

```
# Functions Cell

import scipy
from scipy import linalg

def getSplusSminus(g):
        evs, _ = linalg.eigh(g.am())
        splus = 0
        sminus = 0
        for it in evs:
                if it >0:
                        splus = splus + it^2
                if it<0:
                        sminus = sminus + it^2
        return splus, sminus


# check for all nonbipartite unicylic graphs (nonbip) on n vertices
# outputs: number of graphs
#                       [max s^+ , [list of all graphs attaining that s^+]]
#                       [min s^+ , [list of all graphs attaining that s^+]]
#                       [max s^- , [list of all graphs attaining that s^-]]
#                       [min s^- , [list of all graphs attaining that s^-]]

def getSplusSminusUnicyclicMaxMin(n):
    opties = str(n) + " " + str(n) + " -c"
    g = graphs.nauty_geng(opties)
    splu = {}
    smin = {}
    numgs = 0
    for G in g:
        curr = G.graph6_string()
        if not G.is_bipartite():
            numgs = numgs +1
            nowp, nowm = getSplusSminus(G)
            if round(nowp, 6) in splu.keys():
                splu[round(nowp, 6)].append(curr)
            else:
                splu[round(nowp, 6)] = [curr]
            if round(nowm, 6) in smin.keys():
                smin[round(nowm, 6)].append(curr)
            else:
                smin[round(nowm, 6)] = [curr]
```

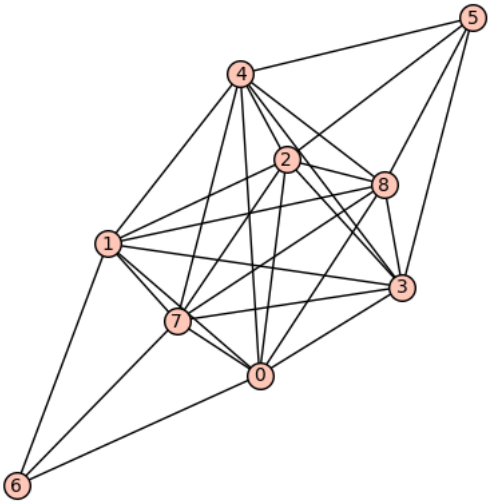
```
    kp = list(splu.keys())
    km = list(smin.keys())
    kp.sort()
    km.sort()
    Pmin = kp[0]
    Pmax = kp[-1]
    Mmin = km[0]
    Mmax = km[-1]
    return numgs, [Pmax, splu[Pmax]], [Pmin, splu[Pmin]], [Mmax, smin[Mmax]], [Mmin, smin[Mmin]]

# for each n, obtains statistics on s^+, s^- and which is bigger
# input: nn the number of vertices (only up to 9)
# output:   number of connected graphs
#           number of bipartite graphs
#           number of graphs where s^+ > s^-
#           number of graphs where s^- > s^+
def SplusMinusStats(nn):
    opties = str(nn) + " -c"
    g = graphs.nauty_geng(opties)
    numgs = 0
    numbip = 0
    pgm = 0
    mgp = 0
    for G in g:
        numgs = numgs +1
        if G.is_bipartite():
            numbip = numbip + 1
        else:
            sp, sm = getSplusSminus(G)
            if sp > sm:
                pgm = pgm + 1
            elif sm > sp:
                mgp = mgp + 1
    return numgs, numbip, pgm, mgp
```

```
# Example 2.6
g=Graph('H~{}Nv|')
show(g)
spG,smG=getSplusSminus(g)
```




```
ge=g.copy()
ge.delete_edge([5,6])
show(ge)
spGe,smGe=getSplusSminus(ge)
print 'Positive Energy of G: ',spG
print 'Positive Energy of G-e: ',spGe
print 's^+(G-1) - s^+(G):', spGe-spG
```

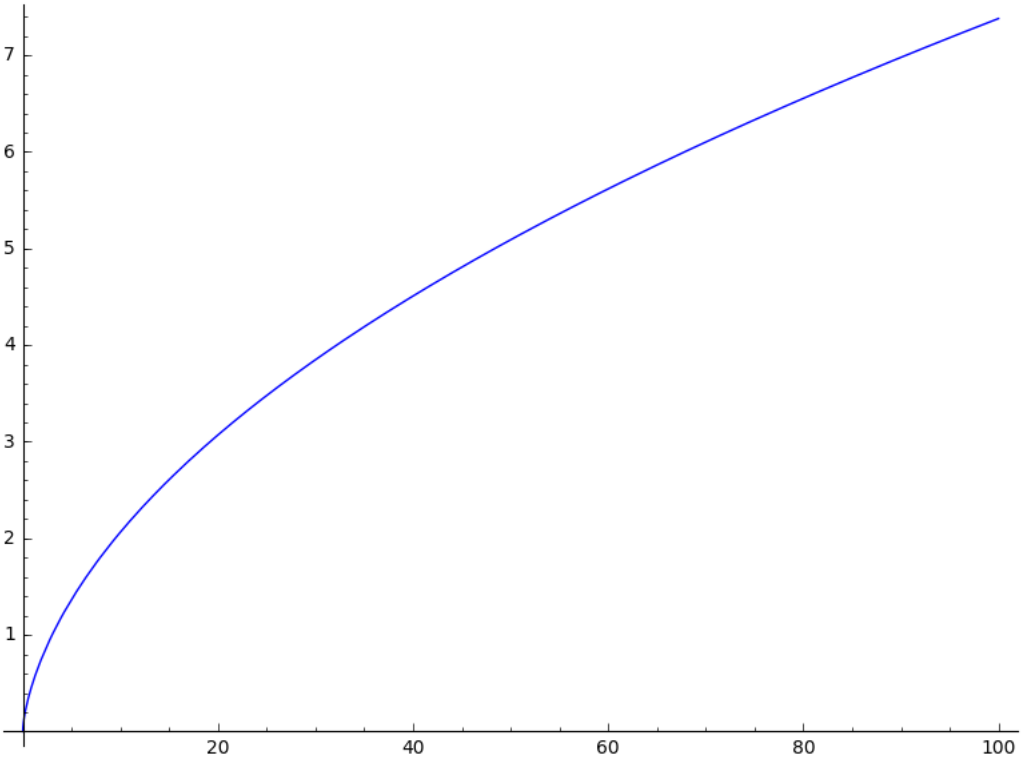
```
Positive Energy of G:  44.8424707858
Positive Energy of G-e:  44.8475921409
s^+(G-1) - s^+(G): 0.00512135504776
```




```
# Plotting the graph in Figure 3.2
```

evaluate

```
plot(pi/(2*arccos((x-1)/(x+1))) - 1/2, (0,100))
```




```
# Verify eigenvalues of H^3_5 for Proposition 3.9
```

```
h35=Graph({0:[1],2:[0,1,3],3:[4]})
show(h35)
a=h35.adjacency_matrix()
a.eigenvalues()
```

```
[1, -1, -1.675130870566646?, -0.5391888728108891?, 2.214319743377535?]
```


```
# Generate Table 4.1, in format sutiable for copy/paste into latex (hence the inclusion of "&")
# The order of the entries is
n, # number graphs, number s^+> s-, number s-^> s^+, % s^- > s+, number bipartite, number s^+=s^-
```

```
for k in [2..8]:
    stats=SplusMinusStats(k)
    perc=numerical_approx(stats[3]/stats[0]*100,digits=6)
    print k, "&",stats[0],"&", stats[2],"&", stats[3],"&", perc,"&", stats[1],"&", stats[0]-stats[2]-stats[3]
```

```
2 & 1 & 0 & 0 & 0.000000 & 1 & 1
3 & 2 & 1 & 0 & 0.000000 & 1 & 1
4 & 6 & 3 & 0 & 0.000000 & 3 & 3
5 & 21 & 15 & 1 & 4.76190 & 5 & 5
6 & 112 & 93 & 2 & 1.78571 & 17 & 17
7 & 853 & 795 & 14 & 1.64127 & 44 & 44
8 & 11117 & 10848 & 87 & 0.782585 & 182 & 182
```

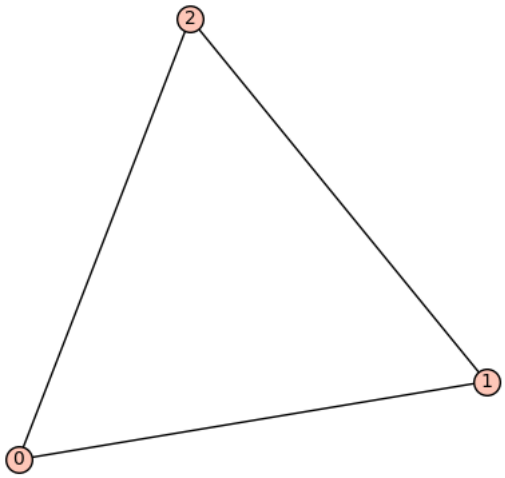
```
# Generate data for Table 4.2, including documenting the graphs
```

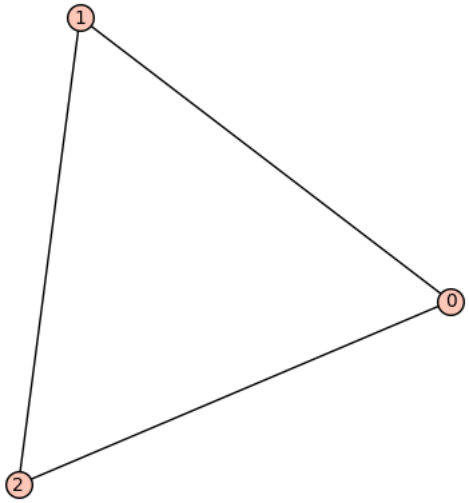
```
# lists of number of graphs, min s^+, min s^- for order = 3,...,18
ng=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
spmin=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
smmin=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
```

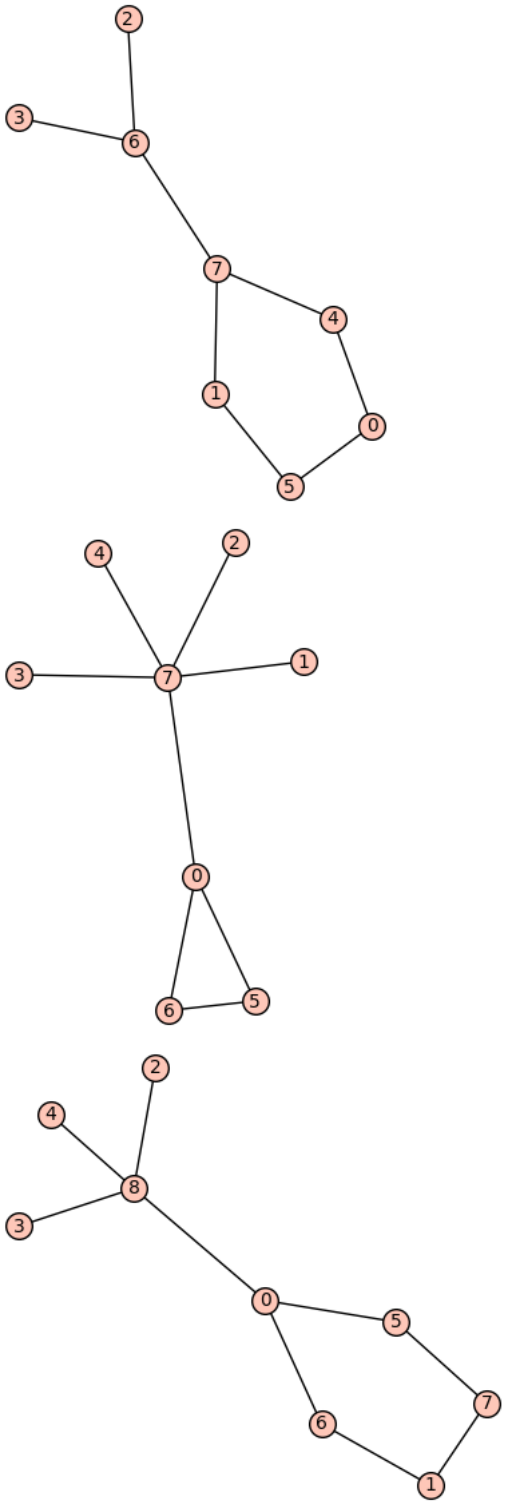
```
for nn in [3..10]:
    data=getSplusSminusUnicyclicMaxMin(nn)
    ng[nn-3]=data[0]
    spmin[nn-3]=data[2][0]
    smmin[nn-3]=data[4][0]
    print nn, "****************************************"
    print data
    gspmin=Graph(data[2][1][0])
    gsmmin=Graph(data[4][1][0])
    show(gspmin)
    show(gsmmin)
```

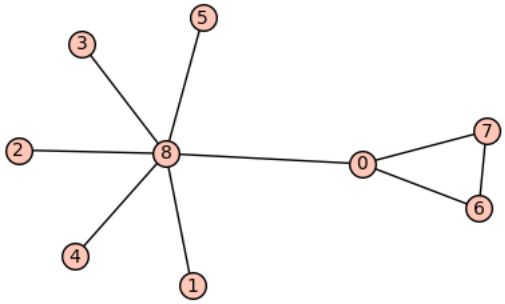
```
3 ****************************************
```

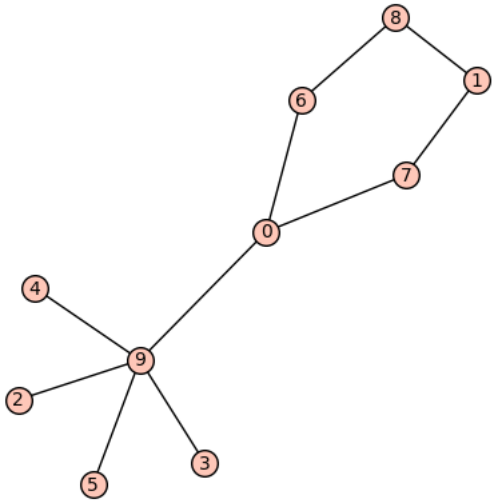
```
(1, [4.0, ['Bw']], [4.0, ['Bw']], [2.0, ['Bw']], [2.0, ['Bw']])
4 ****************************************
(1, [4.806063, ['CV']], [4.806063, ['CV']], [3.193937, ['CV']],
[3.193937, ['CV']])
5 ****************************************
(4, [5.903212, ['DQw']], [4.763932, ['DUW']], [5.236068, ['DUW']],
[4.096788, ['DQw']])
6 ****************************************
(8, [6.926792, ['ECYW']], [5.8548, ['ECpo']], [6.1452, ['ECpo']],
[5.073208, ['ECYW']])
7 ****************************************
(23, [7.939657, ['F?aN_']], [6.797054, ['FCQb_']], [7.202946,
['FCQb_']], [6.060343, ['F?aN_']])
8 ****************************************
(55, [8.948095, ['G?ACNo']], [7.786641, ['G?b@aS']], [8.213359,
['G?b@aS']], [7.051905, ['G?ACNo']])
9 ****************************************
(155, [9.954171, ['H??CCF{']], [8.78153, ['H?AEAIw']], [9.21847,
['H?AEAIw']], [8.045829, ['H??CCF{']])
10 ****************************************
(403, [10.958804, ['I???CA@~_']], [9.778404, ['I??CE@An?']], [10.221596,
['I??CE@An?']], [9.041196, ['I???CA@~_']])
```

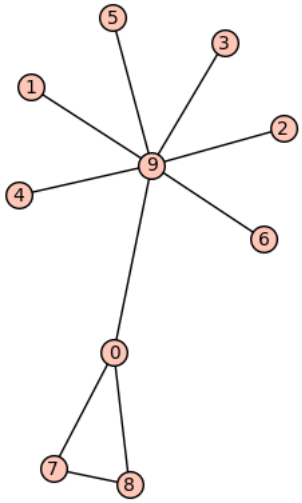


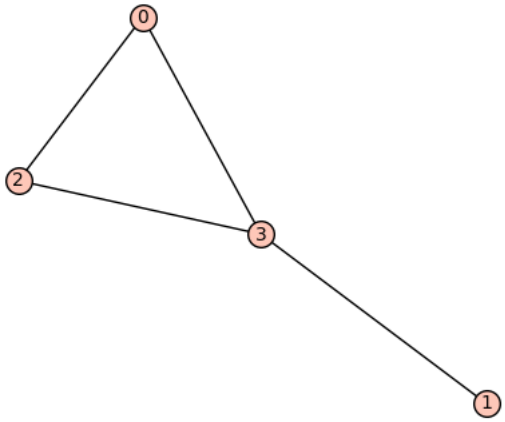


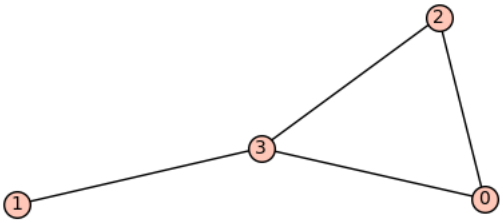

2
3
6
7
4
1
0
5
4
2
3
7
1
0
6
5
2
4
8
3
0
5
7
6
1

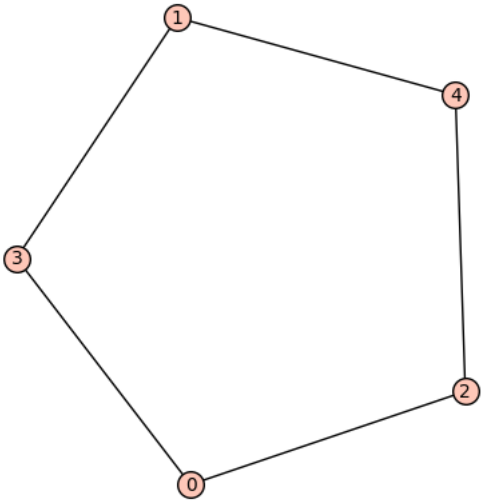

5
3
2
8
0
7
6
4
1

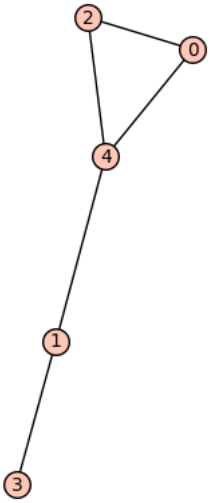

8
1
6
7
0
4
9
2
3
5

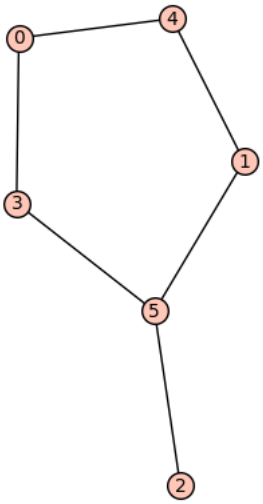

5
3
1
2
9
4
6
0
7
8

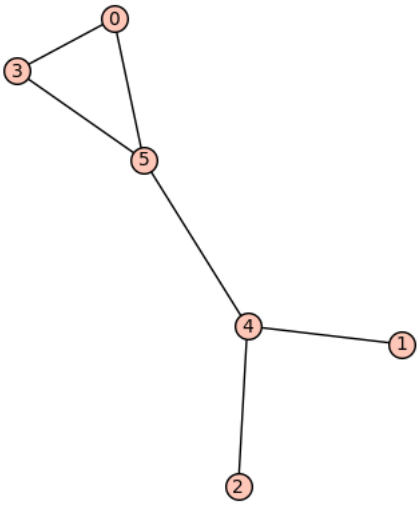

0
2
3
1

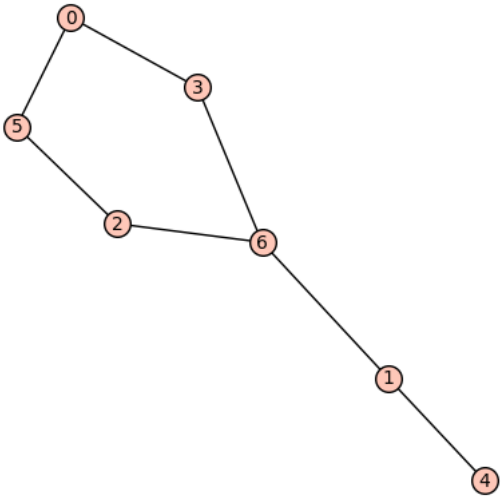

2
3
1
0

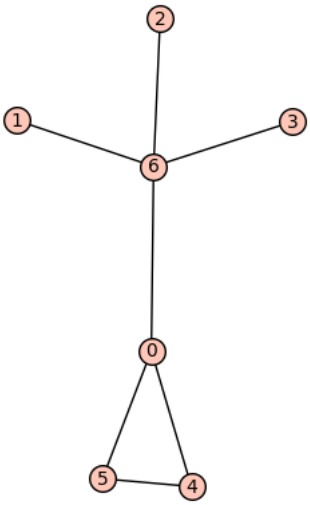

1
4
3
2
0


2
0
4
1
3


4
0
1
3
5
2

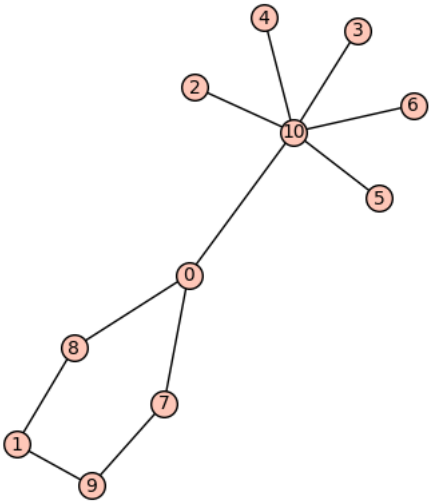

0
3
5
4
1
2

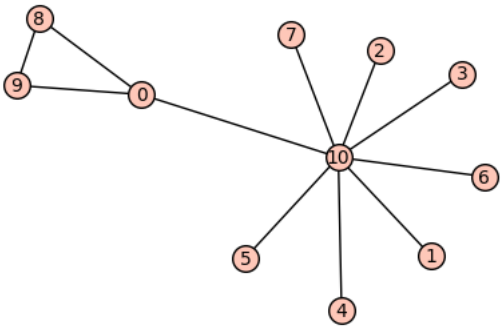

0
3
5
2
6
1
4

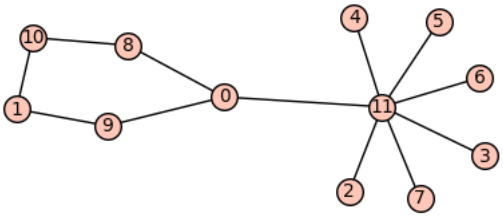

2
1
3
6
0
5
4

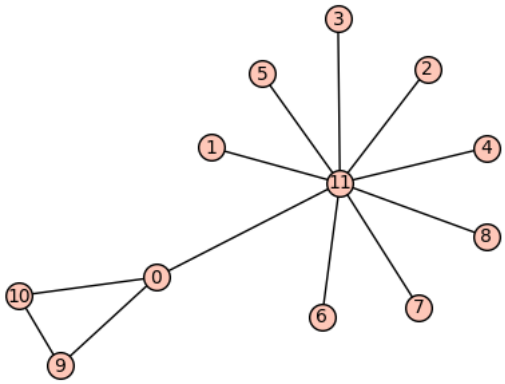
```
for nn in [11..14]:
    data=getSplusSminusUnicyclicMaxMin(nn)
    ng[nn-3]=data[0]
    spmin[nn-3]=data[2][0]
    smmin[nn-3]=data[4][0]
    print nn, "**************************************"
    print data
    gspmin=Graph(data[2][1][0])
    gsmmin=Graph(data[4][1][0])
    show(gspmin)
    show(gsmmin)
```

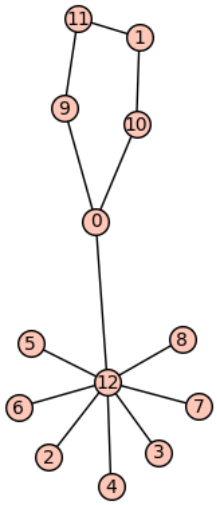
```
11 **************************************
(1116, [11.962479, ['J????A?_N}?']], [10.776269, ['J???CB?OT{?']],
[11.223731, ['J???CB?OT{?']], [10.037521, ['J????A?_N}?']])
12 **************************************
(3029, [12.965481, ['K??????_C?~{']], [11.774708, ['K????A?oA@Vw']],
[12.225292, ['K????A?oA@Vw']], [11.034519, ['K??????_C?~{']])
13 **************************************
(8417, [13.967988, ['L???????C?O@~{']], [12.773512, ['L??????_E?GAnw']],
[13.226488, ['L??????_E?GAnw']], [12.032012, ['L???????C?O@~{']])
14 **************************************
(23285, [14.970118, ['M?????????O?_@~}?']], [13.772564,
['M???????C?W?OAn{?']], [14.227436, ['M???????C?W?OAn{?']], [13.029882,
['M?????????O?_@~}?']])
```

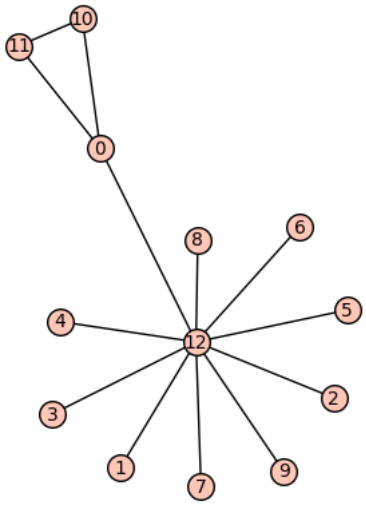


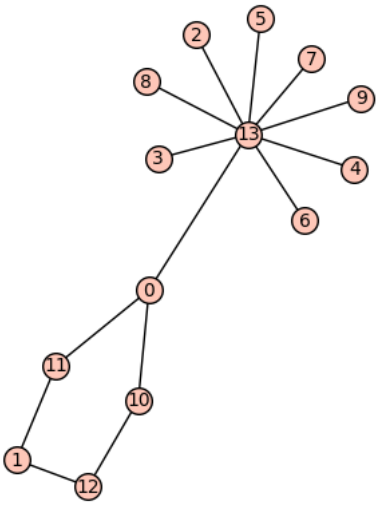


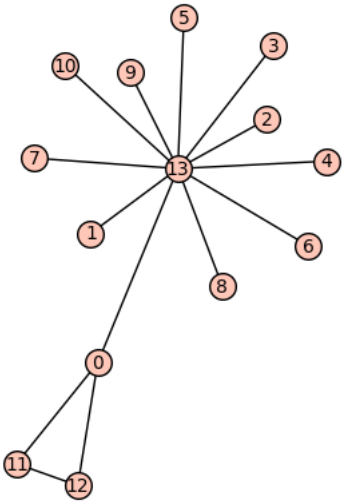



3
5
2
1
4
11
8
0
10
6
7
9


11
1
9
10
0
5
8
12
6
7
2
3
4

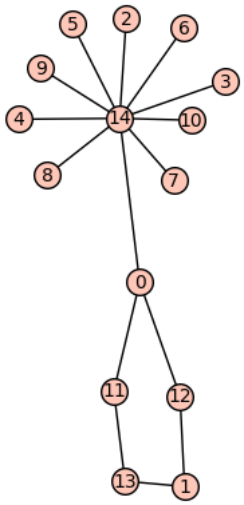

10
11
0
8
6
4
5
12
3
2
1
7
9

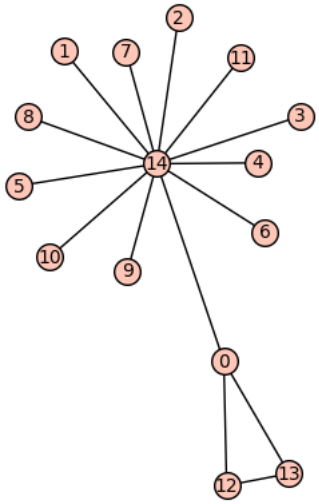





```
nn=15
data=getSplusSminusUnicyclicMaxMin(nn)
ng[nn-3]=data[0]
spmin[nn-3]=data[2][0]
smmin[nn-3]=data[4][0]
print nn, "****************************************"
print data
gspmin=Graph(data[2][1][0])
gsmmin=Graph(data[4][1][0])
show(gspmin)
show(gsmmin)
```

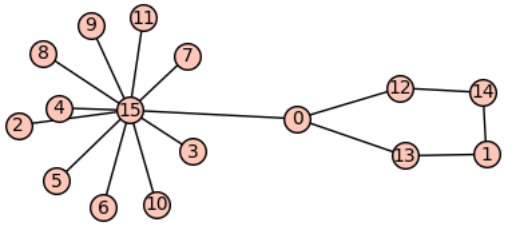
```
15 ****************************************
(65137, [15.971955, ['N???????????_?_?~~_']], [14.771792,
['N??????????O?o?O@V~?']], [15.228208, ['N??????????O?o?O@V~?']],
[14.028045, ['N???????????_?_?~~_']])
```

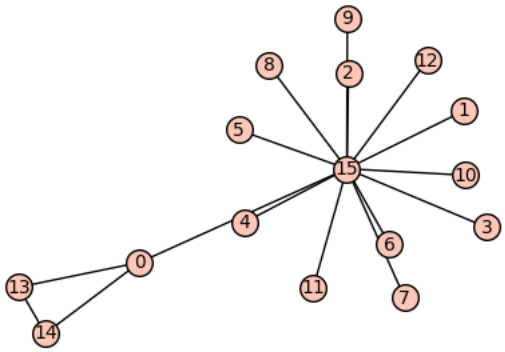





```
nn=16
data=getSplusSminusUnicyclicMaxMin(nn)
ng[nn-3]=data[0]
spmin[nn-3]=data[2][0]
smmin[nn-3]=data[4][0]
print nn, "****************************************"
print data
gspmin=Graph(data[2][1][0])
gsmmin=Graph(data[4][1][0])
show(gspmin)
show(gsmmin)
```

```
16 ****************************************
(182211, [16.973558, ['O????????????_?O?N~{']], [15.771151,
['O???????????_?o?G?T~w']], [16.228849, ['O???????????_?o?G?T~w']],
[15.026442, ['O????????????_?O?N~{']])
```



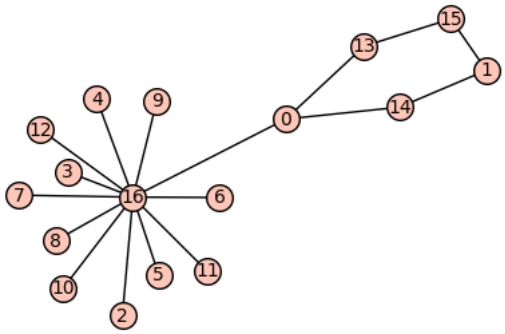


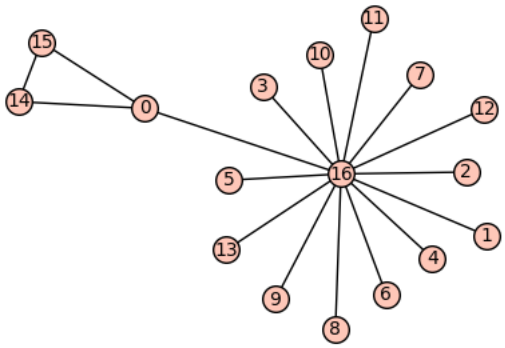

```
Data for n=17 and n=18 was not run on the ISU Sage server but is imported here:
```

```
nn=17
data=getSplusSminusUnicyclicMaxMin(nn)
ng[nn-3]=data[0]
spmin[nn-3]=data[2][0]
smmin[nn-3]=data[4][0]
print nn, "****************************************"
print data
```

```
17 ****************************************
(512625, [17.974971, ['P??????????????O?C?@~~o']], [16.77061,
['P?????????????_?W?A?An~_']], [17.22939, ['P?????????????_?W?A?An~_']],
[16.025029, ['P??????????????O?C?@~~o']])
```

```
show(Graph('P?????????????_?W?A?An~_'))
show(Graph('P??????????????O?C?@~~o'))
```

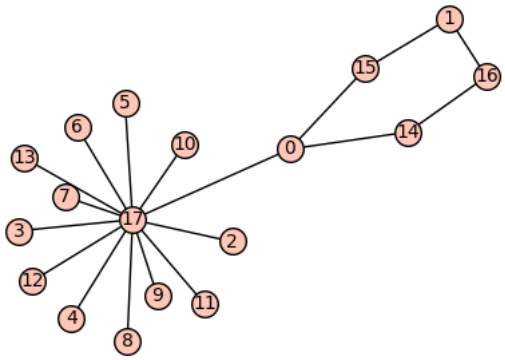


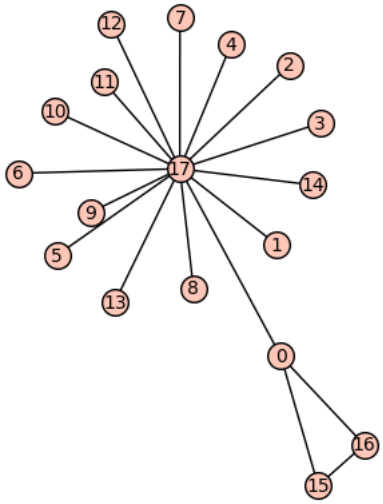


```
nn=18
data=getSplusSminusUnicyclicMaxMin(nn)
ng[nn-3]=data[0]
spmin[nn-3]=data[2][0]
smmin[nn-3]=data[4][0]
print nn, "**************************************"
print data
```

```
18 **************************************
(1444444, [18.976226, ['Q??????????????????C??_?F~~_']], [17.770146,
['Q???????????????O?E??O?I~~?']], [18.229854,
['Q???????????????O?E??O?I~~?']], [17.023774,
['Q??????????????????C??_?F~~_']])
```

```
show(Graph('Q???????????????O?E??O?I~~?'))
show(Graph('Q??????????????????C??_?F~~_'))
```






```
# Output Table 4.2, in format sutiable for copy/paste into latex (hence the inclusion of "&")
```

```
print  "total graphs &", ng[0], "&", ng[1], "&", ng[2], "&", ng[3], "&", ng[4], "&", ng[5], "&", ng[6],"&",
ng[7],"&", ng[8],"&", ng[9], "&", ng[10], "&",ng[11],"&",ng[12],"&",ng[13] "&", ng[14], "&", ng[15]
print  "min $s^+(G)$ &",spmin[0],"&", spmin[1],"&",spmin[2],"&",spmin[3],"&",spmin[4],"&",spmin[5], "&",
spmin[6],"&", spmin[7], "&", spmin[8] , "&",spmin[9],"&",spmin[10],"&",
spmin[11],"&",spmin[12],"&",spmin[13],"&",spmin[14],"&",spmin[15]
print  "min $s^-(G)$ &",smmin[0],"&", smmin[1],"&",smmin[2],"&",smmin[3],"&",smmin[4],"&",smmin[5],"&",
```

```
total graphs & 1 & 1 & 4 & 8 & 23 & 55 & 155
& 403 & 1116 & 3029 & 8417 & 23285 & 65137 &
182211 & 512625 & 1444444
min $s^+(G)$ & 4.0 & 4.806063 & 4.763932 & 5.8548 &
6.797054 & 7.786641 & 8.78153 & 9.778404 & 10.776269
& 11.774708 & 12.773512 & 13.772564 & 14.771792 &
```

```
    15.771151 & 16.77061 & 17.770146
    min $s^-(G)$ & 2.0 & 3.193937 & 4.096788 & 5.073208
    & 6.060343 & 7.051905 & 8.045829 & 9.041196 &
    10.037521 & 11.034519 & 12.032012 & 13.029882 &
    14.028045 & 15.026442 & 16.025029 & 17.023774
```