Python code for bounding the list color function threshold

All of the programs in this supplement are written in Python. Our first program allows us to find the m and n values for which the lower bound in Lemma 13 demonstrates that $P_\ell(K_{2,n}, m) < P(K_{2,n}, m)$.

```
import math

n=1
m=3 #choose any desired value of $m\geq 3$

def PGL_1(x, y, z, w, m):
    return (m-2) * (
        ((m-1) ** (x + y + z * (m-3) / (m-2) + w * (m-4) / (m-2))) *
        (m ** ((z + 2*w) / (m-2)))
    )

def PGL_2(x, y, z, w, m):
    if (m==3):
        return 0
    return (m-2)*(m-3) * (
        ((m-2) ** (x + y + z * (m-4) / (m-2) + w * (m-4)*(m-5) / ((m-2) * (m-3)))) *
        ((m-1) ** (z * 2 / (m-2) + w * 4 * (m-4) / ((m-2) * (m-3)))) *
        (m ** (w * 2 / ((m-2) * (m-3))))
    )

def PGL_3(x, y, z, w, m):
    return 4*(m-2)* (
        ((m-2) ** (x / 2 + y / 2 + z * 3*(m-3) / (4 * (m-2)) + w * (m-4) / (m-2))) *
        ((m-1) ** (x / 2 + y / 2 + z * m / (4 * (m-2)) + w * 2 / (m-2))) *
        (m ** (z / (4 * (m-2))))
    )

def PGL_4(x, y, z, w, m):
    return 4 * (
        ((m-2) ** (y / 4 +z / 2 + w)) *
        ((m-1) ** (x + y / 2 + z / 2)) *
        (m ** (y /4))
    )


# This function is the lower bound for P(G,L) given in the statement of the lemma.
def PGL(x,y,z,w,m):
    return PGL_1(x,y,z,w,m)+PGL_2(x,y,z,w,m)+PGL_3(x,y,z,w,m)+PGL_4(x,y,z,w,m)
```

```
# This function is $P(K_{2,n},m)$.
def PG(n,m):
    return m * ((m-1)**(n)) + m*(m-1) * ((m-2)**(n))

stop = False
chromatic_polynomial = 0
# This loop runs for all $m\geq 4$.
if (m!=3):
    while (n >= 1):
        chromatic_polynomial = PG(n,m)
        for x in range(n+1):
            for y in range(n+1-x):
                for z in range(n+1-x-y):
                    w = n-x-y-z
                    if (chromatic_polynomial > PGL(x,y,z,w,m)):
                        stop = True
        if (not stop):
            print("n = " + str(n) + " is good")
            n += 1
        else:
            print("n = " + str(n) + " is the first bad n")
            break
# This loop runs when m=3. It takes into account the fact that $w$ is
  necessarily equal to $0$ when $m=3$.
if (m==3):
    while (n >= 1):
        chromatic_polynomial = PG(n,m)
        for x in range(n+1):
            for y in range(n+1-x):
                z = n-x-y
                if (chromatic_polynomial > PGL(x,y,z,0,m)):
                    stop = True
        if (not stop):
            print("n = " + str(n) + " is good")
            n += 1
        else:
            print("n = " + str(n) + " is the first bad n")
            break
```

Our second program is needed for the proof of Statement (ii) of Theorem 7.

```
import math

n=1
m=4
```

```
# This function is the lower bound for $P(G,L)$ when $d=1$.
def PGL_d1(x):
    return ((3**(x) * 4**(n-x)) +
    6*(6**(x/2) * 36**((n-x)/3)) +
    9*(15552**(x/9) * 5184**((n-x)/9)))

# This function is the lower bound for $P(G,L)$ when $d=0$.
def PGL_d0():
    return 16*(3**(n/2)*4096**(n/16))

# This function is $P(K_{2,n},m)$.
def PG(n,m):
    return m * ((m-1)**(n)) + m*(m-1) * ((m-2)**(n))

stop = False
chromatic_polynomial = 0
while (n >= 1 and n <= 24):
    chromatic_polynomial = PG(n,m)
    for x in range(n+1):
        if (chromatic_polynomial > PGL_d1(x)
        or chromatic_polynomial > PGL_d0()):
            stop = True
    if (not stop):
        print("n = " + str(n) + " is good")
        n += 1
    else:
        break
```

Our final program is needed for the proof of Statement (iii) of Theorem 7.

```
import math

n=1
m=5

# This function is the lower bound for $P(G,L)$ when $d=2$.
def PGL_d2(x, y, z, w, v):
    return (4*(4**(x) * 4**(y) * 20**(z/2) * 20**(w/2) * 5**(v)) +
    12*(12**(x/2) * 12**(y/2) * 2880**(z/6) * 2880**(w/6) * 5120**(v/6)) +
    9*(4**(x) * 230400**(y/9) * 48**(z/3) * 103680**(w/9) * 36**(v/3)))
```

```
# This function is the lower bound for $P(G,L)$ when $d=1$.
def PGL_d1(x, y, z, w, v):
    return ((4**(x) * 4**(y) * 4**(z) * 5**(w) * 5**(v)) +
    8*(12**(x/2) * 12**(y/2) * 12**(z/2) * 128000**(w/8) * 128000**(v/8)) +
    16*(4**(x) * 3538944000**(y/16) * 240**(z/4) * 192**(w/4) * 34560**(v/8)))

# This function is the lower bound for $P(G,L)$ when $d=0$.
def PGL_d0(x, y, z):
    return 25 * (4**(x) * (3**(4/25*y)*4**(17/25*y)*5**(4/25*y)) *
(3**(6/25*z)*4**(13/25*z)*5**(6/25*z)))

# This function is $P(K_{2,n},m)$.
def PG(n,m):
    return m * ((m-1)**(n)) + m*(m-1) * ((m-2)**(n))

stop = False
chromatic_polynomial = 0
while (n >= 1 and n <= 43):
    chromatic_polynomial = PG(n,m)
    for x in range(n+1):
        for y in range(n+1-x):
            for z in range(n+1-x-y):
                for w in range(n+1-x-y-z):
                    v = n-x-y-z-w
                    if (chromatic_polynomial > PGL_d2(x,y,z,w,v)
                    or chromatic_polynomial > PGL_d1(x,y,z,w,v)):
                        stop = True
    for x in range(n+1):
        for y in range(n+1-x):
            z = n-x-y
            if (chromatic_polynomial > PGL_d0(x,y,z)):
                stop = True
    if (not stop):
        print("n = " + str(n) + " is good")
        n += 1
    else:
        break
```