

SRG-76-30-8-14 eliminating 16-coclique

```
def inters(a,b):#number of common 1s in binary representation
    c=0
    for i in range(16):
        c+=(a%2) * (b%2)
        a=floor(a/2)
        b=floor(b/2)
    return c

def check24(x):
    if x==2 or x==4:
        return 1
    return 0

basec4 =
[ZZ('1111111100000000',base=2),ZZ('1100000011111100',base=2),ZZ('0000111100001111',base=2),ZZ('0011000011110011',base=2)]

goodtobase = []
for i in range(2^16):
    good = True
    if inters(i,2^16-1)==8:
        for j in range(4):
            if (i==basec4[j]) or not(check24(inters(i,basec4[j]))):
                good = False
        if good:
            goodtobase.append(i)

gtb = Graph(Matrix([[check24(inters(i,j)) for j in goodtobase] for i in goodtobase]))

print gtb.order(), gtb.size()

time gtb.clique_number()
906 176672
15
Time: CPU 1002.41 s, Wall: 1591.08 s
```