Pràctica 4 Grafs. Camins i connexió

Molts problemes es poden resoldre modelant l'entorn que els representa mitjançant un graf i viatjant al llarg de les arestes del graf.

Dedicarem aquesta sessió de pràctiques a presentar diferents funcions de SAGE que ens permetran treballar les qüestions bàsiques sobre camins i connexió de grafs.

0. Repàs de conceptes bàsics

Representa el graf de Petersen i respon les preguntes següents:

In [ ]:
```
G1=graphs.PetersenGraph()
show(G1)
```

a) Quins són els camins més curts entre els vèrtexs 1 i 3? I entre els vèrtexs 1 i 9?

b) Quina és la **distància** entre aquests vèrtexs?

c) Quina és la distància màxima entre dos vèrtexs del graf?

d) Quin és el cicle més curt que podem trobar en el graf? Quina és la seva longitud? (Aquesta longitud s'anomena **cintura** del graf (**"girth"** en anglès)

e) És connex el graf?

f) Podem eliminar alguns vèrtexs de G per obtenir un graf que no sigui connex? Quantes components connexes té el nou graf?

g) Pots trobar un circuit eulerià sobre G?

h) Pots trobar un circuit hamiltonià sobre G?

i) Pots trobar arbres generadors de G?

Considera el graf G2 amb conjunt de vèrtexs $V = \{0..10\}$, i conjunt d'arestes

$$E = \{\{0, i\} : 1 \leq i \leq 10\} \cup \{\{2i - 1, 2i\} : 1 \leq i \leq 5\}$$

Defineix el graf G2 en SAGE, representa'l i a continuació respon les qüestions anteriors per a aquest graf.

Considera el graf G3 amb conjunt de vèrtexs $V = \{0, 1, 2, 3, \cdots, 12\}$ i conjunt d'arestes

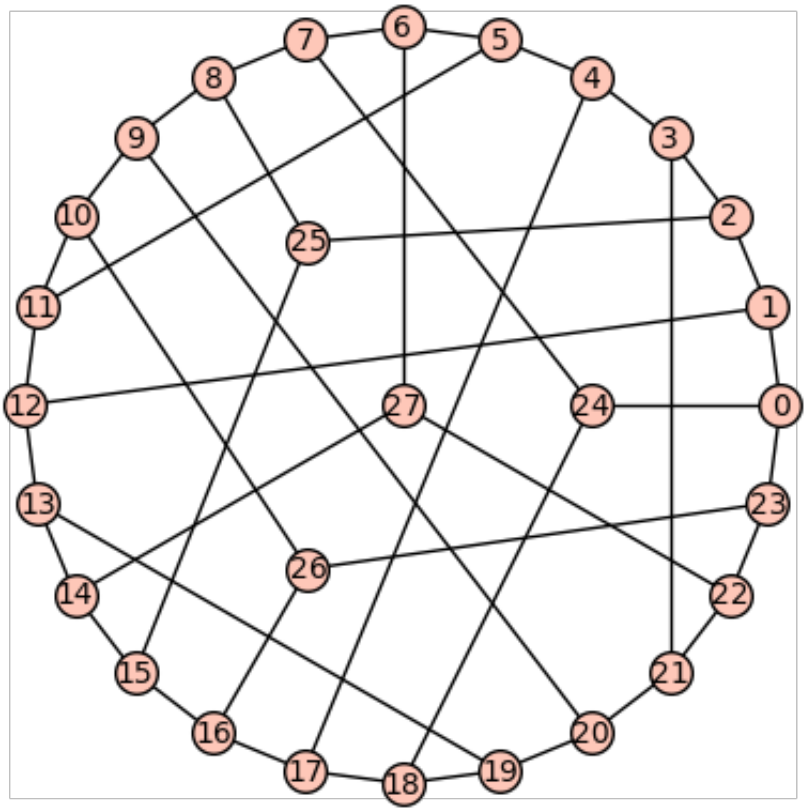
$$E = \{\{0, i\}, \{1, i\} : \ i \in \{2, \cdots, 12\}\}$$

Defineix el graf G3 en SAGE, representa'l i a continuació respon les qüestions anteriors per a aquest graf.

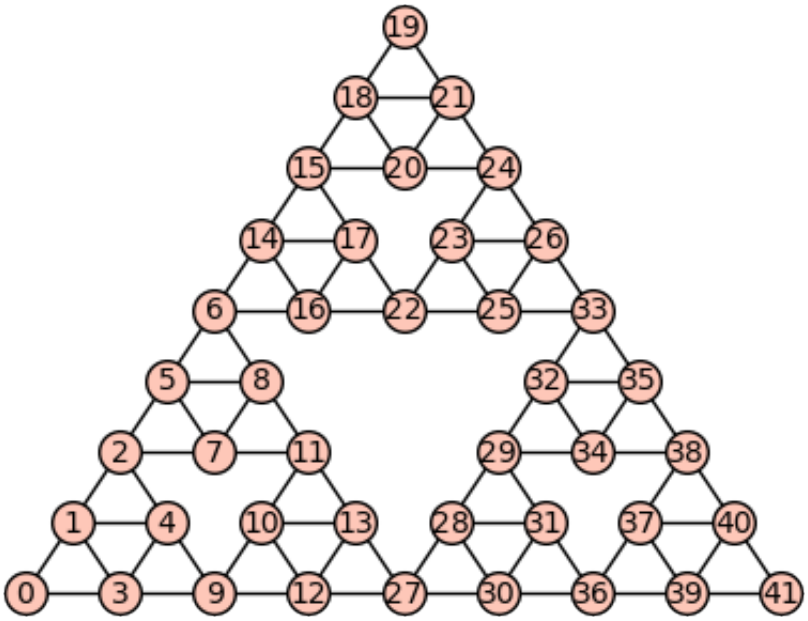
In [ ]:

Has pogut resoldre totes les qüestions plantejades fins al moment per què els grafs que has estudiat son molt petits. En els problemes reals, però, els grafs que intervenen solen ser molt grans, i per això cal disposar d'algoritmes per resoldre aquestes qüestions. SAGE incorpora molts d'aquests algoritmes. A partir d'ara, veurem les comandes de SAGE que els duen a terme. Les aplicarem al graf de Coxeter i un graf de Sierpinski-Gasket.

In [12]:
```
G=graphs.CoxeterGraph()
show(G)
```



In [20]:
```
H=graphs.SierpinskiGasketGraph(4)
show(H)
```



1. Distàncies

La comanda **G.distance(u,v)** calcula la distància que hi ha entre dos vèrtexs u,v del graf G. Troba la distància entre els vèrtexs 1 i 3.

In [ ]:

Utilitza la instrucció **G.shortest_path(1,3)** per a obtenir els vèrtexs interns d'aquest camí més curt.

In [ ]:

També podrem calcular tots els camins entre dos vèrtexs donats utilitzant la instrucció **G.all_paths(u,v)** de SAGE. Aplica-la per a obtenir tots els camins més curts entre els vèrtexs 1 i 3.

In [ ]:

La matriu **distància** d'un graf ens diu la distància que hi ha entre qualsevol parell de vèrtexs del graf. La podem trobar amb la comanda **G.distance_matrix()**. Utilitza-la i llegeix sobre la matriu la distància entre els vèrtexs 2 i 9.

In [ ]:

Calcula amb una instrucció de SAGE el diàmetre del graf de Coxeter.

In [ ]:

La longitud del cicle més curt d'un graf, (la cintura (girth) del graf) es pot calcular amb la instrucció **G.girth()**. Utilitza-la per a calcular el girth del graf de Coxeter.

In [ ]:

Responeu les mateixes qüestions per al graf de Sierpinski-Gasket.

In [ ]:

2. Connectivitat

Troba una instrucció de SAGE que et permeti determinar si el graf de Coxeter és connex.

In [ ]:

Prova la mateixa instrucció amb el graf següent.

In [1]:
```
g=Graph({1:[2,3],2:[3],4:[],5:[6,7,8]})
```

En cas que un graf no sigui connex, pots calcular-ne el nombre de components connexes utilitzant la comanda **g.connected_components_number()**

In [ ]:

S'obté la distribució de vèrtexs del graf per components connexes amb la instrucció **g.connected_components()**

In [ ]:

Representa el graf **g**, per a comprovar que efectivament, els seus vèrtexs es distribueixen d'aquesta forma en components connexes

In [ ]:

Responeu les mateixes qüestions per al graf de Sierpinski-Gasket.

In [ ]:

3. Cicles

Comprova si el graf G és eulerià. En cas afirmatiu troba un circuit eulerià utilitzant **G.eulerian_circuit()**.

In [ ]:

Comprova si és hamiltonià.

In [ ]:

Calcula la vèrtex connectivitat del graf de Coxeter amb **G.vertex_connectivity()**

In [ ]:

Té el graf de Coxeter un vèrtex de tall? En cas afirmatiu elimina'l del graf i calcula el nombre de components connexes del graf resultant.

In [ ]:

Responeu les mateixes qüestions per al graf de Sierpinski-Gasket.

In [ ]:

4. Arbres generadors

Troba un arbre generador del graf G1 aplicant l'algorisme BFS amb arrel 0. Ho pots fer mitjançant la instrucció **list(G1.breadth_first_search(start=0))**. Comprova el resultat que dona SAGE.

In [ ]:

Troba un arbre generador del graf G1 aplicant l'algorisme DFS amb arrel 0. Ho pots fer mitjançant la instrucció **list(G1.depth_first_search(start=0))**. Comprova el resultat que dona SAGE.

In [ ]:

Responeu les mateixes qüestions per al graf de Sierpinski-Gasket.

In [ ]: