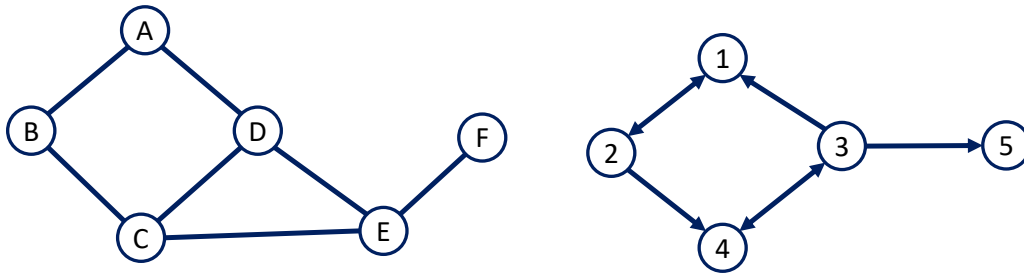


Nous avons vu deux implémentations possibles des graphes. Il existe déjà des modules permettant de manipuler des graphes en Python, comme par exemple le module **networkx**. Soit les deux graphes suivants :



- Tapez le script Python suivant et vérifiez que ce script crée bien les deux graphes donnés en exemple. (Remarque : ne tenez pas compte des avertissements éventuels apparaissant dans la console Python).

```
import networkx as nx          # pour créer des graphes
import matplotlib.pyplot as plt # pour les représenter graphiquement

a = nx.Graph()                # création d'un graphe non orienté
a.add_node("A")                # add_node : méthode pour rajouter un noeud
a.add_node("B")
a.add_node("C")
a.add_node("D")
a.add_node("E")
a.add_node("F")

a.add_edge("A", "B")           # add_edge : ajout d'une arête non orientée
a.add_edge("A", "D")
a.add_edge("B", "C")
a.add_edge("C", "D")
a.add_edge("C", "E")
a.add_edge("D", "E")
a.add_edge("E", "F")

nx.draw_networkx(a, with_labels = True) # trace le graphe avec les labels
plt.show()                          # affiche la fenêtre graphique

b = nx.DiGraph()               # création d'un graphe orienté
b.add_node("1")
b.add_node("2")
b.add_node("3")
b.add_node("4")
b.add_node("5")

b.add_edge("1", "2")           # les arêtes étant orientées, il faut créer
b.add_edge("2", "1")           # les liaisons dans les deux sens, si nécessaire
b.add_edge("2", "4")
b.add_edge("3", "1")
b.add_edge("3", "4")
b.add_edge("3", "5")
b.add_edge("4", "3")

nx.draw_networkx(b, with_labels = True)
plt.show()
```

2025
Bruno PEYROT



- En vous servant de ce module **networkx**, créez et affichez les graphes correspondant aux exemples « Facebook » et « Instagram » du polycop sur les graphes.
- En vous aidant de la documentation du module **networkx**, reproduire les différents algorithmes étudiés dans le polycop sur les algorithmes portant sur les graphes.
- Créez une fonction qui transforme un objet de type **Graphe** d'une des deux classes **Graphe** que nous avons créées précédemment en objet de type **Graph** utilisé par le module **networkx**.
- Pour les plus courageux, implémentez l'algorithme de Dijkstra en Python : on utilisera au choix l'une des deux classes précédentes, ou alors le module **networkx**.