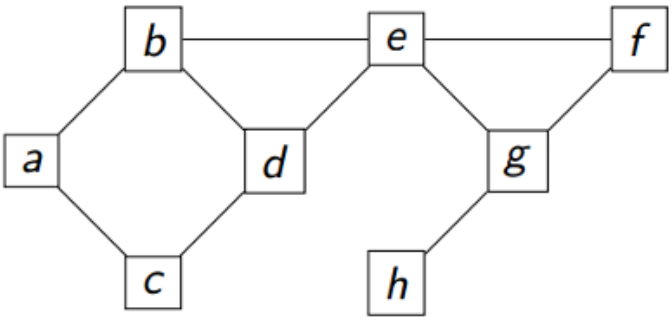
C2 – Exercices d'entrainement en autonomie

Exercice 1 : **Manipulation d'un graphe**

On donne le graphe :



Ecrire un programme python :

Q1. A partir de la représentation du graphe ci-dessous, construire un dictionnaire d'adjacence modélisant le graphe (on pourra vérifier le résultat en utilisant les fonctions d'affichage fournies sur le site).

Q2. A partir du dictionnaire précédemment construit, créer la liste des labels de chaque sommet (« a » sera le label du sommet de numéro 0, et ainsi de suite). Créer aussi la liste d'adjacence.

Q3. A partir du dictionnaire ou de la liste d'adjacence, créer la matrice d'adjacence.

Parcours

Pour cette partie, il est recommandé d'utiliser le dictionnaire d'adjacence.

Q4. En partant du nœud « f », parcourir le graphe en largeur en affichant à chaque étape le label du nœud visité.

Q5. En partant du nœud « f », parcourir le graphe en profondeur en affichant à chaque étape le label du nœud visité.

Plus court chemin :Algorithme de Dijkstra :

Q6.

a. Transformer le graphe en graphe pondéré. Pour cela, on définit un poids, dans la matrice d'adjacence, en remplaçant chaque 1 par une variable aléatoire comprise entre 1 et 10. Attention, le caractère symétrique de la matrice doit être conservé. On pourra utiliser la fonction randint du module random.

b. Appliquer l'algorithme de Dijkstra pour déterminer le chemin le plus court entre les sommets « c » et « f ».