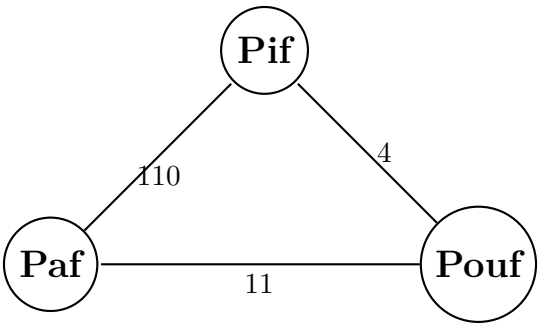
TD 15 Manipulation de graphes

Lycée Louis Thuillier - Informatique - PCSIB -

Pour tout le TD on travaillera sur un graphe représentant les interactions sociales entre des individus.
- ▷ chaque sommet est le nom d'une personne
- ▷ chaque arrête représente les messages envoyés sur le mois, le poids étant le nombre de messages



On remarque alors que Pif et Paf entretienne une conversation régulière et équilibré alors que Pouf est plutôt à l'écart.

Pour manipuler ce dictionnaire on récupérera le fichier *Dictionnaire_Graphe.txt* via le module "json" :

```
import json
with open('Dictionnaire_Graphe.txt', 'r') as file:
    Graphe = json.load(file)
```

On pourra ensuite manipuler le dictionnaire *Graphe* "comme d'habitude", notamment pour tester nos fonctions.

Premières manipulation de graphe

1. Combien y a-t-il de personnes dans le graphe ? Générer la liste de tous les prénoms présent dans le graphe.
2. Écrire une fonction *passage_matrice* qui reçoit un graphe sous la forme d'un dictionnaire et qui renvoie ce même graphe mais représenté par une matrice d'adjacence.
3. Écrire une fonction *degre_sortant* qui prend en argument un graphe sous la forme d'un dictionnaire et un de ses sommets et renvoie le degré sortant de ce sommet. Même question pour le degré entrant.
4. Combien de messages Jennifer (🔥🔥🔥 ***Attention !*** *pas de majuscule dans le graphe !*) a-t-elle envoyé ? Et reçu ? Qui est son/sa meilleur ami/amie ? Est-elle ami avec Pierre ?

Théorème des poignée de mains et Barack Obama

Selon la théorie des six degré de séparation, toute personne célèbre est à six poignée de mains de vous. Théorie notamment illustré avec Barack Obama, ancien président des États-Unis.
- ▷ vous êtes à une poignée de main de quelqu'un si il fait partie de vos connexions (*vous avez échanger un ou plusieurs messages*)
- ▷ vous êtes à deux poignée de main de quelqu'un si vous êtes à une poignée de main d'une personne à une poignée de main de lui (*i.e.* il fait partie des amis de vos amis)

▷ etc ...

Dans un graphe cela se représente par un chemin : deux personnes sont à n poignée de mains l'une de l'autre s'il existe au minimum un chemin de n arrêtes les reliant.

5. Écrire une fonction *Existence_chemin* qui prend en entrée un graphe représenté par un dictionnaire et une liste de sommets du graphe et renvoie $True$ si le chemin défini par la liste des sommets existe et $False$ sinon.
6. Créer la liste de toutes les personnes à une poignée de mains de Pierre.
7. Créer la liste de toutes les personnes à deux poignée de mains de Pierre

Mais comment tester l'existence d'un chemin entre deux sommets sans connaître les sommets intermédiaires par lesquels passer ? On pourrait tester "au pif" tous les chemins possibles mais on imagine que cela prendrait un temps infiniment long. Il convient alors d'avoir une méthode efficace et systématique pour se "promener" sur un graphe depuis un sommet : ce sont les parcours en largeur et en profondeur.