

TP 2 – Files, Familles, Fret

1. ÉCHAUFFEMENT

- (0) Pour bien comprendre la différence entre piles et files, essayons de construire les fonctions *peek* et *vide*.

2. PARCOURS DE GRAPHES

On considère un ensemble de personnes reliées ou non par des liens de parenté. En l'occurrence, voici une partie de l'arbre généalogique de Louis XIV et de son épouse Marie-Thérèse d'Autriche. Comme vous pouvez le voir il y a une très grande parentalité entre les époux (ce qui pourrait expliquer les 5 enfant morts dans l'enfance sur 6 mis au monde).

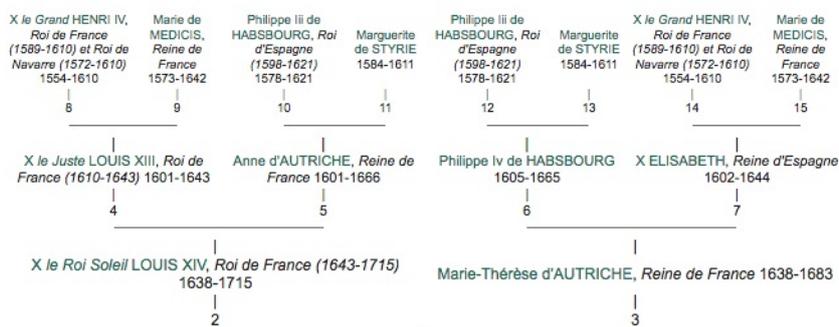
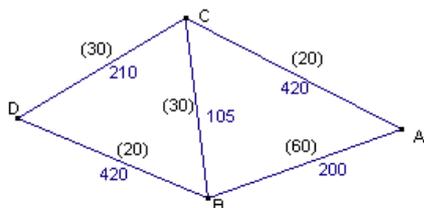


FIGURE 1. Extrait de l'arbre généalogique des Bourbon.

- (1) Représenter cet arbre généalogique sous la forme d'un graphe, et déterminer comment le représenter en Python.
- (2) Ecrire un programme qui permette de trouver le chemin le plus court entre deux personnes appartenant au même arbre (l'utilisation des files n'est pas indispensable, bien que très utile).
- (3) Modifier ce programme pour qu'il renseigne en plus le nombre de chemins différents de longueur minimale entre deux personnes.

3. OPTIMISATION DE TRANSPORT

On considère un entrepreneur soucieux de livrer une cargaison d'une ville à une autre. Voici le graphe représentant les connexions entre les villes de transit.



- (4) Écrire un programme qui détermine le chemin le plus court entre les villes A et D.
- (5) Modifier ce programme afin de déterminer le chemin le moins cher.
- (6) Expliquer pourquoi il est nécessaire qu'un tel graphe ne présente aucune boucle de chemin de longueur (en distance ou en péage) négative pour que vos programmes fonctionnent.