Mathématiques & Informatique - F. Gaunard
`http://frederic.gaunard.com`
Lycée Voltaire, Paris 11e.

3 Travaux Pratiques : Dictionnaires et graphes

L'objectif de ce **TP** est de manipuler l'outil dictionnaire de `Python`, notamment dans le parcours de graphe, avec un exercice très classique (qui existe dans une certaine version à l'oral de **Math II**) autour du théorème des quatre couleurs. Les deux activités proposées sont indépendantes.

1 Amuse-bouche : un peu de géographie

1.1 Jointures de dictionnaires

À partir de plusieurs dictionnaires liés entre eux, il est possible d'en créer d'autres pour faire apparaître des informations souhaitées. On appelle cela « *effectuer une jointure* ».
C'est une opération qui peut aussi être effectuée sur des bases de données, dans le Chapitre 5 ainsi que dans le **TP** associé.

1. **a.** Récupérer le fichier `countries.csv`.
   b. L'accès à un fichier `.csv` en `Python` suit à peu près le même principe que l'accès à un fichier `.txt`. En implémentant les lignes suivantes, créer un dictionnaire des capitales.

```
import csv
fichier = open('countries.csv', 'r') # ou open('chemin\countries.csv', 'r')
    # si le fichier n?est pas dans votre répertoire courant
lecture = csv.reader(fichier)
capitale = {ligne[1]: ligne[2] for ligne in lecture}
print(capitale['France'])
```

2. Créer de la même façon, un dictionnaire continent associant les continents aux pays et un dictionnaire monnaie associant les monnaies aux pays.

3. Afficher la liste des noms des pays de l'Océanie, puis de ceux dont la monnaie est l'euro.

4. Afficher la liste des noms des pays non-européens dont la monnaie est l'euro.

5. Créer un dictionnaire associant à chaque continent la liste des monnaies qui y sont utilisées.

1.2 Devinettes

6. Reprendre le dictionnaire `capitale` des capitales de l'exercice précédent.
   Écrire un programme dans lequel l'ordinateur choisit aléatoirement un nom de pays et vous fait deviner sa capitale. En

cas de succès, il vous répond `'Gagné !'`, sinon `'Perdu ! La bonne réponse était...'`.
On utilisera les commandes suivantes:

```
import random as rd
rd.choice(list(capitale.keys()))
```

7. Modifier le programme précédent pour qu'il compte 1 point par succès et affiche votre score final au bout de cinq parties.

8. Modifier le programme précédent pour qu'il vous autorise trois essais à chaque partie.

2 Coloriage en quatre couleurs

Le théorème des 4 couleurs est un théorème classique de théorie des graphes. D'après Wikipedia :

Le théorème des quatre couleurs indique qu'il est possible, en n'utilisant que quatre couleurs différentes, de colorier n'importe quelle carte découpée en régions connexes, de sorte que deux régions adjacentes (ou limitrophes), c'est-à-dire ayant toutes une frontière (et non simplement un point) en commun reçoivent toujours deux couleurs distinctes. Même si l'énoncé de ce théorème est élémentaire, on n'en connaît pas de preuve simple. Les démonstrations connues décomposent le problème en un nombre de sous-cas tellement important qu'elles nécessitent l'assistance d'un ordinateur pour être vérifiées.

L'objectif de cette activité est de trouver un coloriage en 4 couleurs de la carte des régions de France métropolitaine.

La carte des régions de France est alors assimilé à un graphe non orienté dont les sommets sont les régions et deux sommets sont voisins si les régions sont adjacentes.

1. Implémenter ce graphe que l'on nommera `regions` (ou faire un copier/coller de ce script).

Le principe de l'algorithme et le suivant :

- ✗ les couleurs seront représentées par des entiers numérotés à partir de 0;
- ✗ on prend un sommet du graphe au hasard, on regarde les couleurs déjà données à ses voisins, et on lui donnera comme couleur la plus petite valeur non-affectée à un de ses voisins.

2. Écrire une fonction `min_couleur_dispo` prenant trois arguments :
- ✗ `G` : un dictionnaire, représentant le graphe
- ✗ `couleurs` : un dictionnaire, représentant des sommets déjà coloriés
- ✗ `v` : une valeur, représentant un sommet du graphe

et qui renvoie la valeur de couleur la plus petite non déjà utilisée dans `couleur` pour colorier les voisins de `v` dans `G`.

3. En reprenant l'algorithme de parcours en largeur d'un graphe, écrire une fonction `coloriage_graphe` qui prend en argument un graphe et une région de départ et renvoie un *tuple* contenant :

- ✗ le nombre de couleurs utilisées ;
- ✗ un dictionnaire affectant à chaque région sa couleur.

4. Pouvez-vous donner un coloriage utilisant exactement 4 couleurs ?