3 Travaux Pratiques : Dictionnaires et graphes

Exercice 1. **Coloriage en 4 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 cet exercice 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 ? Quid de la carte des 48 états américains ?

Exercice 2. **Correction automatique**

Un enseignant propose des QCM à ses élèves. La solution de ce QCM est représentée par un dictionnaire `solution` dont les clés sont des entiers (représentant les questions) de 1 à 20 et les valeurs un caractère.
Une bonne réponse rapporte un point, une mauvaise réponse enlève un quart de point. Une absence de réponse n'ajoute ni n'enlève aucun point. La note finale est un nombre entre 0 et 20.

Ainsi, chaque *copie* rendue est aussi représentée par un dictionnaire définie de la même manière que précédemment sauf que si l'élève ne répond pas à une question, la valeur correspondante n'apparait pas dans le dictionnaire.
Enfin, l'ensemble des copies est un nouveau dictionnaire `paquet` dont les clés sont les étudiants (représentés par des chaînes de caractère) et les valeurs les dictionnaires modélisant leurs copies.

Écrire une fonction `correction` qui prend en argument les dictionnaires `solution` et `paquet` et renvoie un dictionnaire qui associe chaque étudiant à sa note.

Exercice 3. **Suite de Syracuse**

Pour tout entier $p \in \mathbb{N}^*$, on définit la suite de Syracuse associée par $u_0 = p$ et $\forall n \in \mathbb{N}, u_{n+1} = \begin{cases} u_n/2 & \text{si } u_n \text{ pair} \\ 3u_n + 1 & \text{sinon} \end{cases}$.

On admet que pour toute valeur raisonnable de p (disons $p \leq 5 \times 2^{60}$), la suite finit par boucler sur $1, 4, 2$.
On note $f(p)$ le plus petit n tel que $u_n = 1$.

1. Quelle est la plus grande valeur de $f(p)$ pour $1 \leq p \leq 10000$?
2. Combien d'entiers $p \leq 10^5$ vérifient $f(p) \leq 30$?

Exercice 4. **Secret Santa**

Écrire une fonction `secret_santa` qui prend en argument une liste d'élève `L` (dont les éléments sont donc des chaînes de caractères) et qui renvoie un dictionnaire donc les clés sont chacun des élèves et les valeurs les noms de l'élève choisi au hasard à qui la clé fait son unique cadeau.
Chaque élève fait et reçoit un unique cadeau ; l'application qui va de l'ensemble des élèves dans lui-même correspondant à l'association est donc bijective et on ne se fait pas de cadeau à soi même (elle n'a donc aucun élément invariant).
La commande `randint(a, b)` du package `numpy.random` qu'on importera sous l'alias `rd`) renvoie un entier choisi au hasard avec équiprobabilité entre a et $b - 1$.