TP D'INFORMATIQUE 3

Systèmes de vote et dictionnaires

L'objectif de ce TP est d'écrire en Python différents systèmes de votes, qui prennent en entrée les votes des électeurs et renvoient un candidat élu. Nous utiliserons à cet effet la structure de dictionnaire.

1 Scrutin à un tour

Dans le scrutin à un tour, des votants expriment chacun un choix parmi une liste de candidat, et le candidat remportant le plus de suffrages remporte l'élection.

On choisit d'utiliser un **dictionnaire** pour compter le score de chaque candidat. Un dictionnaire est un objet associant des **valeurs** à des **clés**, ici un nombre de votes à un nom de candidat. Par exemple, si la liste des votes est ['Alice', 'Bob', 'Alice', 'Dave', 'Alice', 'Charlie', 'Bob'], le dictionnaire résultant sera {'Alice': 3, 'Bob': 2, 'Dave': 1, 'Charlie': 1}.

On rappelle qu'on initialise un dictionnaire vide D avec $D = \{\}$, qu'on teste si une clé c appartient à D avec $c \text{ in } D$, qu'on accède à la valeur associée à la clé c dans D avec $D[c]$, et qu'on associe la valeur v à la clé c dans D avec $D[c] = v$ (ce qui ajoute la clé c dans le dictionnaire si elle n'y était pas déjà).

1. Écrire une fonction `creer_dico_resultats` prenant en argument une liste de chaînes de caractères et renvoyant le dictionnaire résultant, sur le modèle de l'exemple ci-dessus.
2. Écrire une fonction `cle_de_valeur_max` prenant en argument un dictionnaire supposé non vide et renvoyant une clé associée à la valeur maximale. Sur le dictionnaire de l'exemple précédent, la fonction doit donc renvoyer 'Alice'. On pourra supposer que les valeurs du dictionnaire sont des entiers positifs. On peut parcourir les clés c d'un dictionnaire D avec une boucle `for c in D`:
3. En déduire une fonction `scrutin_un_tour` prenant en argument une liste de votes et renvoyant le vainqueur du scrutin à un tour correspondant.

2 Scrutin à deux tours

Le scrutin à deux tours est par exemple utilisé pour l'élection présidentielle française. À l'issue d'un premier tour, deux candidats en tête des votes sont conservés pour un second tour où ils s'affrontent en duel.

On peut déterminer les votes du second tour à partir de la liste de préférences totale de chaque votant. On reprend et complète l'exemple précédent en ces 7 listes de préférences :

Alice > Charlie > Dave > Bob	Bob > Charlie > Dave > Alice	Alice > Dave > Charlie > Bob
Dave > Charlie > Bob > Alice	Alice > Bob > Charlie > Dave	Charlie > Dave > Bob > Alice
Bob > Dave > Charlie > Alice		

1. Quel candidat serait élu avec ce système de vote ?
2. Écrire une fonction `candidats_second_tour` prenant en argument le dictionnaire des résultats du premier tour et renvoyant le couple des deux candidats retenus au second tour.
Sur le dictionnaire {'Alice': 3, 'Bob': 2, 'Dave': 1, 'Charlie': 1}, la fonction devrait ainsi renvoyer 'Alice', 'Bob'. Le traitement d'éventuels cas d'ex-aequo est laissé libre.
3. Écrire une fonction `vote_second_tour` prenant en argument une liste classant les candidats, et les deux candidats retenus au premier tour, et renvoyant le candidat retenu qui apparaît en premier dans cette liste. Cette fonction permet d'obtenir le vote au second tour d'un votant à partir de sa liste de préférences.
Ainsi, `vote_second_tour(['Charlie', 'Dave', 'Bob', 'Alice'], 'Alice', 'Bob')` doit renvoyer 'Bob'.
4. En déduire une fonction `scrutin_deux_tours` prenant en argument la liste des listes de préférences des votants, et renvoyant le vainqueur du scrutin à deux tours correspondant. Ainsi :

```
scrutin_deux_tours([ ['Alice', 'Charlie', 'Dave', 'Bob'],
                    ['Bob', 'Charlie', 'Dave', 'Alice'], ['Alice', 'Dave', 'Charlie', 'Bob'],
                    ['Dave', 'Charlie', 'Bob', 'Alice'], ['Alice', 'Bob', 'Charlie', 'Dave'],
                    ['Charlie', 'Dave', 'Bob', 'Alice'], ['Bob', 'Dave', 'Charlie', 'Alice'] ])
```

doit renvoyer le résultat déterminé à la question 1.

5. On dit qu'un système de vote est sujet au vote utile si des votants peuvent avoir intérêt à voter pour un autre candidat que celui qu'ils préfèrent réellement. Est-ce le cas du scrutin à un tour? Du scrutin à deux tours?

3 Dictature aléatoire

Dans ce mode de scrutin, les votants donnent seulement leur choix préféré, puis un bulletin est tiré au hasard et désigne le vainqueur de l'élection.

1. Implémenter ce mode de scrutin. On pourra utiliser `random` avec `from random import *` pour charger la bibliothèque `random`, et l'appel `randint(a,b)` pour tirer un entier uniformément entre les entiers a et b (inclus).
2. Ce mode de scrutin est-il sujet au dilemme du vote utile?

4 Scrutin par élimination

À chaque tour, le candidat avec le moins de voix est éliminé (en cas d'égalité, on décidera ici d'éliminer parmi les ex-aequo le premier candidat dans l'ordre alphabétique). Le dernier candidat encore en lice remporte alors l'élection.

Comme pour le scrutin à deux tours, on prendra en entrée la liste des préférences totales de chaque votant.

1. En reprenant l'exemple de la partie 2, quel candidat serait élu avec ce système de vote?
2. Écrire une fonction `liste_sans` prenant en argument une liste l et un élément a et renvoyant la liste obtenue à partir de l en supprimant les occurrences de a .
3. Écrire une fonction `enlever_candidat` prenant en argument une liste de listes de préférences L et un candidat p , et renvoyant la liste obtenue à partir de L en enlevant toutes les occurrences de p . Ainsi :

```
enlever_candidat([['Bob', 'Dave', 'Charlie'], ['Charlie', 'Dave', 'Bob']], 'Dave')
```

doit renvoyer `[['Bob', 'Charlie'], ['Charlie', 'Bob']]`.
4. Écrire une fonction `cle_de_valeur_min` prenant en argument un dictionnaire non vide et renvoyant une clé associée à la valeur minimale. On pourra utiliser `float('inf')` qui joue le rôle de $+\infty$ en python. Si plusieurs clés sont associées à la valeur minimale, on renverra la plus petite de ces clés pour l'ordre alphabétique (on peut comparer deux chaînes de caractères avec `<`).
5. Écrire une fonction `dernier` prenant en argument le tableau actuel de listes de préférences et renvoyant le candidat éliminé ce tour-ci.
Ainsi, `dernier([['Bob', 'Dave', 'Charlie'], ['Charlie', 'Dave', 'Bob']])` doit renvoyer `'Dave'`.
6. En déduire une fonction `scrutin_elimination` implémentant ce système de vote.
7. Ce système de vote est-il sujet au dilemme du vote utile?

5 Scrutin de Condorcet

Un vainqueur de Condorcet est un candidat qui bat n'importe quel autre candidat en duel. On dit qu'un mode de scrutin vérifie le principe de Condorcet s'il élit le vainqueur de Condorcet dès que celui-ci existe.

1. Les scrutins précédents vérifient-ils le principe de Condorcet?
2. Donner un exemple de vote sans vainqueur de Condorcet.
3. Écrire une fonction `scrutin_condorcet` prenant en argument le tableau des listes de préférences des votants et renvoyant le vainqueur de Condorcet s'il existe, et `None` sinon.