

L2 MIASHS – Informatique S5

TD2 : Structures de données

2019-2020

Ensembles :

Un ensemble en Python correspond à la notion mathématique d'ensemble. C'est une structure de données dans laquelle les éléments ne sont pas ordonnés, et où il ne peut pas y avoir de doublon.

- `s=set()` – création d'un ensemble vide,
- `s=set(e)` – création d'un ensemble reprenant les éléments de `e`, fonctionne avec liste, chaîne de caractères...
- `s.add(val)` – Ajout d'un élément à l'ensemble `s`
- `s.update([val1, val2])` – Ajout de plusieurs éléments,
- `if val in s:` – Recherche d'un élément `val` dans `s`,
- `for x in s:` – Parcours des éléments de `s`,
- `s.remove(val)` – Retrait d'un élément, produit une erreur si `val` introuvable,
- `s.discard(val)` – Retrait d'un élément, aucune erreur, même si `val` introuvable.

Dictionnaires :

Un dictionnaire est une structure de donnée associative : chaque valeur est associée à une clé. On accède à une entrée du dictionnaire en y faisant référence avec la clé.

- `d={}` – création d'un dictionnaire vide.
- `d[cle]=valeur` – affectation de `valeur` à l'entrée `cle` du dictionnaire. Si l'entrée `cle` n'existait pas dans le dictionnaire, elle est créée.
- `d[cle]` – permet d'accéder à la valeur associée à la clé dans le dictionnaire.
- `for x in d.keys() / d.values()` – `x` parcourt les clés, respectivement les valeurs présentes dans `d`.
- `for x,y in d.items()` – simultanément, `x` parcourt les clés, et `y` les valeurs.
- `del d[cle]` – retire l'entrée du dictionnaire (et donc aussi la valeur associée).
- `if x in d` – avec `d` un dictionnaire, teste si `x` est *une clé* dans `d`

Exercice 1

Que fait le programme suivant ? Expliquez les affichages.

```
1 print("dictionnaires")
2 abso = {-7:7, 3:3, -0.25:0.25}
3 for key, value in abso.items ():
4     print(key, value)
5 print(-0.25 in abso.values (), 0.25 in abso.values ())
6 print(-0.25 in abso, 0.25 in abso)
7 for key, value in enumerate (abso):
8     print(key , value)
9
10 print("ensembles")
11 ens = set(abso) # prend les clés
12 ens.update([3,100])
13 for x in ens :
14     print(x)
15 if 100 in ens :
16     ens.remove(100)
17 ens.discard(100)
18 print(ens)
19 ens.add(3)
20 ens.add(3)
21 ens.remove(3)
22 print(ens)
```

Exercice 2

Proposez une fonction `fusion(d1,d2)` qui renvoie la fusion de deux dictionnaires sans modifier aucun des deux.

Par exemple si `a = {1 : 1, 2 : 3, 5 : 1}` et `b = {0 : 3, 6 : 3, 2 : 1}`, le résultat attendu de `fusion(a,b)` est `{1 : 1, 2 : 1, 5 : 1, 0 : 3, 6 : 3}`.

Exercice 3

Ecrivez une fonction qui prend comme paramètre une liste `x` et renvoie un dictionnaire indiquant pour chaque élément de la liste son nombre d'apparitions dans celle-ci.

Les clés du dictionnaire sont les éléments de la liste, et la valeur associée à chaque élément `x` est son nombre d'apparitions dans la liste.

Par exemple si `x = [2, 5, 2, 2, 1]`, elle renvoie `{1 : 1, 2 : 3, 5 : 1}`.

Exercice 4

Ecrivez une fonction qui prend en entrée un dictionnaire et un ensemble et vérifie que toutes les valeurs du dictionnaire sont des éléments de l'ensemble.

Exercice 5

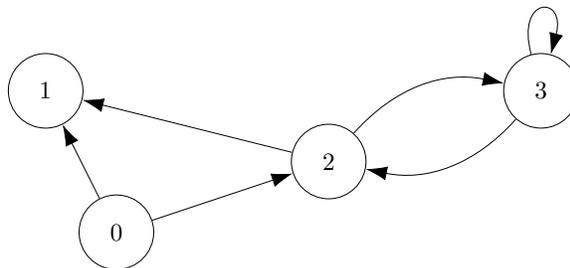
Ecrivez une fonction qui vérifie que deux listes contiennent exactement les mêmes éléments.

Exercice 6 – Graphe – Optionnel

En informatique, le plus souvent, on définit un graphe orienté comme une paire (V, E) avec

- V un ensemble fini d'entiers, appelés *sommets*,
- $E \subseteq V \times V$ un ensemble de couple de sommets appelés *arrêtes*.

Exemple : $G = (\{0, 1, 2, 3\}, \{(0, 1), (2, 1), (0, 2), (2, 3), (3, 2), (3, 3)\})$



G possède donc 4 sommets et 6 arrêtes.

On suppose d'abord qu'on représente un graphe par un couple de deux ensembles, S et V . Ici $g = (\{0, 1, 2, 3\}, \{(0, 1), (2, 1), (0, 2), (2, 3), (3, 2), (3, 3)\})$.

Question 1 Proposez une fonction qui prend un paramètre g et vérifie que g représente bien un graphe. C'est à dire que tous les éléments de $g[1]$ sont des couples constitués d'éléments de $g[0]$.

Un chemin est une suite de sommets $[s_0, s_1, \dots, s_p]$, $s_i \in V$, $(s_i, s_{i+1}) \in E$.

Question 2 On dit que la suite de sommets $s = [s_0, s_1, \dots, s_p]$ est un cycle du graphe $G = (V, E)$ si s vérifie :

- $\forall i, 0 \leq i \leq p, s_i \in V$
- $\forall i, 0 \leq i < p, (s_i, s_{i+1}) \in E$
- $s_0 = s_p$

Proposez une fonction qui prend en paramètre une liste s et un graphe g et vérifie que s est un cycle de g .

Question 3 Proposez une fonction qui à partir d'un graphe g retourne le dictionnaire d tel que :

- les clés sont les sommets,
- la valeur associée à une clé s est l'ensemble des sommets t tel qu'il existe une arrête de s à t .

Pour g , la fonction retourne : $\{0: \{1, 2\}, 1: \text{set}(), 2: \{1, 3\}, 3: \{2, 3\}\}$.

Question 4 Proposez une fonction qui prend en entrée un dictionnaire qui représente un graphe (sous la forme construite à la question précédente) et construit un dictionnaire $d2$ tel que :

- les clés sont les sommets,

- la valeur associée à une clé s est l'ensemble des sommets u tel qu'il existe un sommet t , une arête de t à u et une autre de t à s .

Pour g , la fonction retourne $\{0: \{1, 2\}, 1: \text{set}(), 2: \{1, 3\}, 3: \{2, 3\}\}$

Question 5 L'ensemble des sommets accessibles depuis un sommets $0 \in V$ est l'ensemble des t tel qu'il existe un chemin de s à t .

Proposez une fonction qui construit un dictionnaire tel que :

- les clés sont les sommets,
- la valeur associée à une clé s est l'ensemble des sommets accessibles depuis ce sommet.

Question 6 Un graphe est dit hamiltonien si il existe un cycle qui passe par tous les sommets une et une seule fois. Proposez une fonction qui détermine si un graphe est hamiltonien.