Informatique du Tronc Commun
TD n°2 Les dictionnaires

I Utilisation des dictionnaires

Exercice n°1 Min-Max dans un dictionnaire

Écrire une fonction `min_max` qui prend en argument une liste de nombres non vide et renvoie un dictionnaire dont les clés sont les chaînes `"min"` et `"max"` avec pour valeurs respectives le minimum et le maximum des nombres de la liste.
Par exemple : `min_max([8,5,9,3,1,7])` renverra `{"min":1,"max":9}`.

Exercice n°2 Comparaison de listes

On utilise un dictionnaire pour comparer des listes de nombres qui sont tous de même type, soit du type `int` soit du type `float`.

Q1. Écrire une fonction `occurrences(L:list)->dict` qui prend en argument une liste de nombres et renvoie un dictionnaire dont les clés sont les différents nombres de la liste avec pour valeur le nombre d'occurrences de chaque nombre.

Par exemple `occurrences([3,5,-2,3,3,-2])` renvoie `{-2:2, 3:3, 5:1}`.

Q2. Écrire une fonction `taille(d:dict)->int` qui prend en paramètre un dictionnaire obtenu comme ci-dessus et renvoie la longueur de la liste qui a été considérée.

Q3. Écrire une fonction `compare` qui prend en paramètres deux listes de nombres de même longueur et renvoie `True` si les deux listes contiennent les mêmes nombres, pas nécessairement dans le même ordre, et `False` sinon. On utilisera la fonction `occurrences`.

Quelle est la complexité en temps de cette fonction ?

Exercice n°3 Température

On dispose de noms de villes avec les températures moyennes relevées à une date donnée. Les données sont enregistrées dans une liste de listes comme `[["Paris",12],["Lyon",14],["Marseille",21],...]` On note n la longueur de la liste.

Q1. On souhaite accéder à une donnée, la modifier, ou en ajouter une. Quelle est en fonction de n la complexité en temps de chacune de ces opérations ?

Q2. Afin de trouver la température d'une ville, on trie cette liste, à l'aide d'un algorithme de tri rapide (rappel : il est de complexité $O(n\log_2(n))$), suivant les noms des villes en utilisant l'ordre lexicographique. Puis on effectue une recherche dichotomique du nom de la ville. Quelle est la complexité de cette recherche de température ?

On souhaite stocker ces données dans un dictionnaire dont les clés sont les noms des villes et les valeurs les températures.

Q3. Écrire une fonction `convert(L:list[list])->dict` prenant en paramètre une liste comme ci-dessus et renvoyant le dictionnaire correspondant.

Q4. Écrire ensuite une fonction `temperature(d:dict,ville:str)->float` qui prend en argument un tel dictionnaire et un nom de ville et renvoie la température correspondant à la ville.

Q5. Quelle est la complexité en temps d'une recherche ou d'une modification de température ?

Exercice n°4 Inversion

Un graphe orienté est représenté par un dictionnaire. Les clés sont les sommets du graphe. La valeur associée à une clé `s` est la liste des sommets extrémités des arêtes partant de `s`. On souhaite créer le graphe obtenu à partir d'un graphe initial en inversant le sens des arêtes.

Écrire une fonction `inverse(G:dict)->dict` qui prend un graphe (représenté par le dictionnaire d'adjacence) en paramètre et renvoie le graphe obtenu par l'inversion du sens des arêtes.

Par exemple :

```
>>> G = {'a':['b','c','e'] , 'b':['d'] , 'c':['e'] , 'd':['c','e'] , 'e':[]}
>>> inverse(G)
{'a':[] , 'b':['a'] , 'c':['a','d'] , 'd':['b'] , 'e':['a','c','d'] }
```

Exercice n°5 Matrice

On s'intéresse ici aux matrices parcimonieuses, c'est-à-dire dont la plupart des coefficients sont nuls.

Une telle matrice M de dimensions (n, p) pourra être codée par un dictionnaire ayant pour couples clefs/valeurs :

- `'dim' : (n,p)` , qui donne les dimensions de la matrice ;
- `(i,j) : M_{ij}` , qui donne pour chaque couple (i, j), la valeur de l'élément $M_{i,j} \neq 0$ de la matrice.

Q1. Donner le dictionnaire qui code la matrice : $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 \end{pmatrix}$

Q2. Proposer une fonction d'addition `somme_mat(M1:dict,M2:dict)->dict` de deux matrices parcimonieuses.

Q3. Si la première matrice contient c coefficients non nuls, et la seconde c', quelle est la complexité temporelle de cet algorithme ?

II Dictionnaire et table de hachage

Exercice n°6 Double hachage

Le double hachage est l'une des meilleures méthodes connues pour l'adressage ouvert. Il utilise une fonction de hachage de la forme :

$$h : \begin{array}{ccc} \mathbb{N} \times \mathbb{N} & \longrightarrow & [\![0, m-1]\!] \\ (k, i) & \mapsto & \Big(h_1(k) + i \times h_2(k)\Big) \bmod m \end{array}$$

où h_1 et h_2 sont des fonctions de hachages.
i prend les valeurs suivantes :

- Par défaut, $i = 0$.
- S'il y a collision, on incrémente i de 1, jusqu'à ne plus avoir de collision pour la clé considérée.
- Il reprend la valeur 0 pour la clé suivantes …

Q1. Insérer les clés : 5, 28, 19, 15, 20, 33, 12, 17, 10 dans un tableau de taille $m = 13$ avec $h_1(k) = k \bmod 13$ et $h_2(k) = 1 + \big(k \bmod 12\big)$.

Q2. Proposer une fonction en `Python` qui prend en argument une clé `c` (entier) et la taille `m` de la table, et renvoie la valeur de $h(c)$.

Exercice n°7 Polynôme

On considère des polynômes non nuls à coefficients entiers de degré quelconque mais qui ne contiennent pas plus de cinq monômes. On utilise un tableau de longueur $16 = 8 \times 2$ pour stocker les couples (degré, coefficient) dans lequel on pourrait stocker au maximum huit couples. Les places non occupées contiennent la valeur -1.
La fonction de hachage h est la fonction identité : pour tout $n \in \mathbb{N}$, $h(n) = n$. Donc à un degré qui vaut 10, on associe le nombre 10, soit $h(10) = 10$.
Ensuite, on écrit le degré (la clé), suivi du coefficient(la valeur) à l'indice $(10 \bmod 8) = 2$.
Par exemple, le polynôme $8 + 3x^{10} - 5x^{12}$ est stocké dans un tableau de la forme :

indice 0	0	8
indice 1	-1	-1
indice 2	10	3
indice 3	-1	-1
indice 4	12	-5
indice ...	...	...

Q1. Donner le tableau correspondant au stockage du polynôme $2x^5 - 3x^{34} + 4x^{105}$.

Q2. Quel est le problème avec par exemple le polynôme $8 - 5x^2 + 3x^{10}$?

Q3. En cas de collision, on décide d'utiliser la première place libre suivante. Les monômes sont entrés dans le tableau suivant l'ordre de lecture. Donner un exemple de polynôme de degré minimum qui génère une collision pour chaque monôme excepté le premier.

Q4. On envisage une autre possibilité de stockage avec deux tableaux, un tableau pour les couples (degré, coefficient) et un tableau pour les indices, les deux tableaux ayant pour capacité 8.
Avec le polynôme $4x^3 - 2x^5 + 4x^9$, on obtient les deux tableaux de la manière suivante :

- dans le premier tableau, on écrit chaque degré avec le coefficient correspondant suivant l'ordre des degrés et on complète le tableau avec des 0 ;
- dans le second tableau, on calcule $(d \bmod 8)$ où d est un degré et on place à l'indice trouvé l'indice où on trouve le couple (degré,coefficient) dans le premier tableau. On complète le tableau avec des -1. Extraits des tableaux :

indice 0	3	4
indice 1	5	-2
indice 2	9	4
indice 3	0	0
indice ...	...	...

indice 0	-1
indice 1	2
indice 2	-1
indice 3	0
indice 4	-1
indice 5	1

Donner les deux tableaux correspondant au stockage du polynôme $3x^5 - x^{18} + 7x^{20}$.