

Langage Python

TD 2 : structures de données

Hubert Godfroy

05 novembre 2015

1 Mémoire

Question 1 :

Tester les opérateurs `==` et `is`. Utiliser `id` pour connaître l'emplacement en mémoire d'un objet.

2 Chaînes de caractères

Le mot u est un facteur du mot v si il existe v_1, v_2 tels que $v = v_1uv_2$.

Question 2 :

Construire une fonction testant si un mot est facteur d'un autre

Question 3 :

Construire une fonction permettant de remplacer chaque occurrence d'un facteur par un autre mot.

3 Algorithmes sur les tableaux

Question 4 :

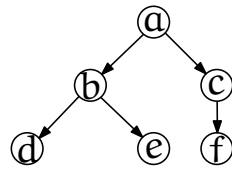
Faire la somme des éléments d'un tableau.

Question 5 :

Étant donnée une liste d'entiers, proposer un algorithme de tri. Que changer si on choisit un tuple d'entier ?

4 Arborescence

Un arbre est une structure de cette forme :



Question 6 :

Proposer une représentation en Python.

Question 7 :

Coder un programme calculant la taille de l'arbre (son plus long chemin de la racine à une feuille).

Question 8 :

Coder une fonction calculant le nombre de nœuds d'un arbre.

5 Gestion des fichiers

Question 9 :

Lire un fichier de mots français et les entrer dans une structure d'ensemble.

6 Dictionnaires

Question 10 :

Construire la liste des mots de taille inférieure à 4.

Question 11 :

Coder une fonction testant si deux mots sont des anagrammes l'un de l'autre.

Question 12 :

Coder une fonction construisant la liste des anagramme d'un mot.

Question 13 :

Construire un dictionnaire associant à chaque mot la liste de ses anagrammes.

Question 14 :

Étant donné une liste de lettres (avec répétition), construire l'ensemble des mots que l'on peut faire avec ces lettres.

7 Distance entre deux mots (difficile)

On peut mesurer la *distance* entre deux mots en cherchant le nombre minimal d'opération de suppression ou d'ajout nécessaire à l'égalisation des deux mots.

Question 15 :

Écrire une fonction calculant cette distance.