
Chapitre 2 : Listes, dictionnaires, tris

4 Exercices

Exemple 4.1 (Matrices et dictionnaires).

1. Comment peut-on manipuler des matrices en Python en utilisant un dictionnaire?

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)
- (i, j) : M_{ij} pour chaque couple (i, j) tel que $M_{i,j} \neq 0$.

2. Donner le dictionnaire qui code la matrice : $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 \end{pmatrix}$.
3. Proposer une fonction d'addition de deux matrices de même dimension.
4. Si la première matrice contient c coefficients non nuls et la seconde c' , quelle est la complexité temporelle de cet algorithme?

Exemple 4.2 (Tri par dénombrement). Soit N un entier naturel non nul. On cherche à trier une liste L d'entiers naturels strictement inférieurs à N .

1. Écrire une fonction `comptage`, d'arguments L et N , renvoyant une liste P dont le k -ième élément désigne le nombre d'occurrences de l'entier k dans la liste L .
2. Utiliser la liste P pour en déduire une fonction `tri`, d'arguments L et N , renvoyant la liste L triée dans l'ordre croissant.
3. Quelle est la complexité temporelle de cet algorithme? La comparer à la complexité d'un tri par insertion ou d'un tri fusion.