

Chapitre 2 : Corrigés des exercices

4 Exercices

Exemple 4.1 (Matrices et dictionnaires).

1. On peut considérer comme clé un 2-uplet qui indique la position du coefficient indiqué comme valeur de la clé. On code également les dimensions sous les clés 'dim'.
2. On peut prendre :
 $M = \{ \text{'dim': (2, 4)}, (1, 2) : 4 \}$
3. On somme les clés correspondantes :

```
def somme_mat ( M1 , M2 ) :
    assert M1 ['dim']== M2 ['dim'] , " les deux matrices doivent être de même dimension "
    M ={ 'dim': M1 ['dim']}
    for c1 in M1 :
        if c1 !='dim':
            if c1 in M2 :
                M [ c1 ]= M1 [ c1 ]+ M2 [ c1 ]
            else :
                M [ c1 ]= M1 [ c1 ]
    for c2 in M2 :
        if c2 !='dim':
            if c2 not in M :
                M [ c2 ]= M2 [ c2 ]

    return M
```

4. Ceci donne un $O(c + c')$ en considérant que les accès aux clés sont à coût constant. Pour une matrice écrite avec une liste de liste c'est en $O(np)$ (deux boucles for imbriquées).

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. On incrémente à l'endroit de l'élément considéré :

```
def comptage(liste,N):
    # Initialisation du tableau de comptage à 0
    tabComptage = [0]*N
    # Création du tableau de comptage
    for k in liste:
        tabComptage[k]=tabComptage[k]+1
    return(tabComptage)
```

2. On utilise le code précédent pour créer ensuite des morceaux de liste de valeurs identiques qu'on concatène :

```
def TriComptage(liste,N):
    tabComptage = [0]*N
    for k in liste:
        tabComptage[k]=tabComptage[k]+1
    tabTrie=[]
    #on crée les blocs de valeurs identiques
    for i in range(N+1):
        for j in range(tabComptage[i]):
            tabTrie.append(i)
    return tabTrie
```

3. Ce tri est linéaire en temps (la boucle sur l'entier j ne donne que N exécutions au total. Ce tri peut sembler plus efficace que le tri fusion ou le tri rapide mais il nécessite une grande mémoire.