

Nom :

Devoir 26

Dans toute la suite, on considère des graphes non pondérés et non orientés. Les sommets sont représentés par des entiers ainsi matrice d'adjacence, liste des listes d'adjacence et dictionnaire des listes d'adjacence sont tels que :

```
#matrice d'adjacence
m=[[0,1,0,1],
   [1,0,1,1],
   [0,1,0,1],
   [1,1,1,0]]
#dictionnaire associée
dico={0: [1, 3], 1: [0, 2, 3], 2: [1, 3], 3: [0, 1, 2]}
#liste des listes d'adjacence
liste=[[0, [1, 3]], [1, [0, 2, 3]], [2, [1, 3]], [3, [0, 1, 2]]]
```

1) Ecrire une fonction `dict_to_list` qui prend en argument un dictionnaire des listes d'adjacence et qui renvoie la liste des listes d'adjacence associée.

2) Ecrire une fonction `mat_to_dict` qui prend en argument une matrice d'adjacence et qui renvoie le dictionnaire des listes d'adjacence associé.

Nom :

Devoir 25

Dans toute la suite, on considère des graphes non pondérés et non orientés. Les sommets sont représentés par des entiers ainsi matrice d'adjacence, liste des listes d'adjacence et dictionnaire des listes d'adjacence sont tels que :

```
#matrice d'adjacence
m=[[0,1,0,1],
   [1,0,1,1],
   [0,1,0,1],
   [1,1,1,0]]
#dictionnaire associée
dico={0: [1, 3], 1: [0, 2, 3], 2: [1, 3], 3: [0, 1, 2]}
#liste des listes d'adjacence
liste=[[0, [1, 3]], [1, [0, 2, 3]], [2, [1, 3]], [3, [0, 1, 2]]]
```

1) Ecrire une fonction `dict_to_list` qui prend en argument un dictionnaire des listes d'adjacence et qui renvoie la liste des listes d'adjacence associée.

2) Ecrire une fonction `mat_to_dict` qui prend en argument une matrice d'adjacence et qui renvoie le dictionnaire des listes d'adjacence associé.