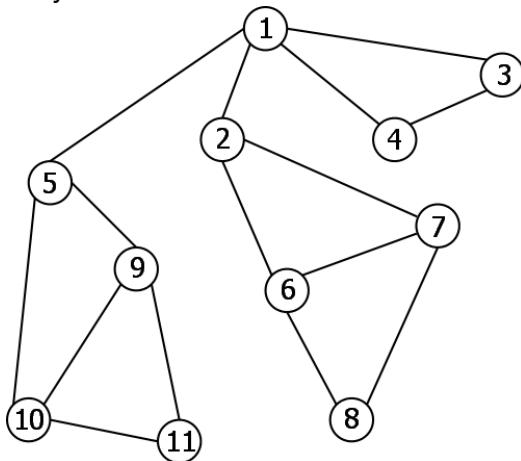


Type de document	Exercices	Classe	Tle	Durée	2h	Date	21/01/2025
Thème et contenu(s)	Graphes						
Capacités attendues	Savoir représenter un graphe formellement mais aussi en Python – Parcours BFS/DFS et calcul de distances						
Prérequis	Arbres, piles, files						
Description	Exercices						

## I) Représentation

Représenter le graphe suivant au moyen d'un dictionnaire contenant la liste des voisins :



## II) Parcours

Testez vos algorithmes de parcours BFS et DFS et vérifiez leur cohérence, par exemple :

DFS récursif depuis 1 {1: True, 2: True, 6: True, 7: True, 8: True, 3: True, 4: True, 5: True, 9: True, 10: True, 11: True}

DFS itératif depuis 1 [1, 2, 6, 7, 8, 3, 4, 5, 9, 10, 11]

BFS 1 depuis 1 [1, 2, 3, 4, 5, 6, 7, 9, 10, 8, 11]

DFS récursif depuis 5 {5: True, 1: True, 2: True, 6: True, 7: True, 8: True, 3: True, 4: True, 9: True, 10: True, 11: True}

DFS itératif depuis 5 [5, 1, 2, 6, 7, 8, 3, 4, 9, 10, 11]

BFS 1 depuis 5 [5, 1, 9, 10, 2, 3, 4, 11, 6, 7, 8]

## III) Code

- Créer une fonction qui prend en paramètre une matrice d'adjacence représentant un graphe (les sommets sont donc des clefs numériques) et renvoie le dictionnaire correspondant. Par exemple :

```
mat = [[0, 1, 0], [0, 0, 1], [0, 1, 1], [1, 0, 1, 1]]
m = matToDic(mat)
renverra :{0: [1, 2], 1: [2], 2: [1, 2], 3: [0, 2, 3]}
```

- Modifier judicieusement l'algorithme du BFS de sorte qu'il renvoie un dictionnaire des distances par rapport au sommet initial, par exemple ce code pourra renvoyer :

```
BFS Distances depuis 1 {1: 0, 2: 1, 3: 1, 4: 1, 5: 1, 6: 2, 7: 2, 9: 2, 10: 2, 8: 3, 11: 3}
```

```
BFS Distances depuis 11 {11: 0, 9: 1, 10: 1, 5: 2, 1: 3, 2: 4, 3: 4, 4: 4, 6: 5, 7: 5, 8: 6}
```