



Sujet

DS07

Graphe

Durée de l'épreuve : 01h55



Correction

Aucun document n'est autorisé.

L'usage de la calculatrice n'est pas autorisé.

Le candidat répond sur feuilles doubles numérotées et garde l'énoncé.

Les traces de recherche, même incomplètes ou infructueuses, seront valorisées.

La qualité de la rédaction, la clarté et la précision des raisonnements seront prises en compte.

Exercice 1 - Un exemple (5 points)

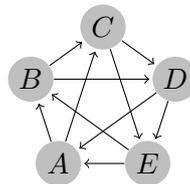


FIGURE 1 – Graphe 1

1. Donner la représentation du graphe 1 par une matrice d'adjacence.
2. Donner la représentation du graphe 1 par une liste d'adjacence.
3. Lister tous les parcours en profondeur sur le graphe 1.
4. Lister tous les parcours en largeur sur le graphe 1.
5. Lister tous les cycles du graphe 1.

Exercice 2 - L'implémentation d'un graphe (5 points)

Implémenter une structure de graphe non orienté simple en paradigme objet avec une liste d'adjacence :

```

1 class Graphe:
2     '''classe modélisant un graphe non orienté simple avec une liste d'adjacence'''
3
4     def __init__(self):
5         '''Constructeur de la classe'''
6
7     def ordre(self):
8         '''Retourne l'ordre du graphe'''
9
10    def taille(self):
11        '''Retourne la taille du graphe'''
12
13    def estVide(self):
14        '''Retourne True si le graphe est vide, False sinon'''
15
16    def ajouteSommet(self, A):
17        '''Ajoute, si nécessaire, le sommet A au graphe'''
18
19    def ajouteArete(self, A, B):
20        '''Ajoute l'arête A-B au graphe (ainsi que les sommets éventuellement)'''

```

Exercice 3 - L'existence d'un chemin (5 points)

1. Implémenter une méthode de parcours en profondeur minimale (uniquement pour permettre de vérifier l'existence d'un chemin entre deux sommets) :

```
22 def parcoursProfondeur(self, A, .....):
23     '''Retourne un parcours en profondeur à partir du sommet A'''
```

2. Implémenter une méthode qui vérifie l'existence d'un chemin entre deux sommets du graphe :

```
25 def existeChemin(self, A, B):
26     '''Retourne True s'il existe un chemin entre A et B'''
```

Exercice 4 - La détermination d'un chemin (5 points)

1. Implémenter une méthode de parcours en profondeur (pour permettre cette fois de déterminer un chemin entre deux sommets) :

```
22 def parcoursProfondeur(self, A, .....):
23     '''Retourne un parcours en profondeur à partir du sommet A'''
```

2. Implémenter une méthode qui détermine un chemin entre deux sommets :

```
28 def chemin(self, A, B):
29     '''Retourne un chemin entre A et B s'il en existe un, None sinon'''
```

Exercice bonus (optionnel)

Implémenter un algorithme qui résout un sudoku à travers l'utilisation d'un graphe.

On considérera chaque cellule comme un sommet du graphe et chaque couple de cellules de la même ligne ou de la même colonne ou de la même région relié par une arête.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8	3				1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

FIGURE 2 – Une grille 9×9 de sudoku