

**Exercice 1 :**

## 1. Matrice d'adjacence

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```

1 g = [[0, 1, 1, 1, 0, 0],
2       [1, 0, 0, 1, 0, 0],
3       [1, 0, 0, 0, 1, 0],
4       [1, 1, 0, 0, 0, 0],
5       [0, 0, 1, 1, 0, 0],
6       [0, 0, 0, 0, 0, 0]]

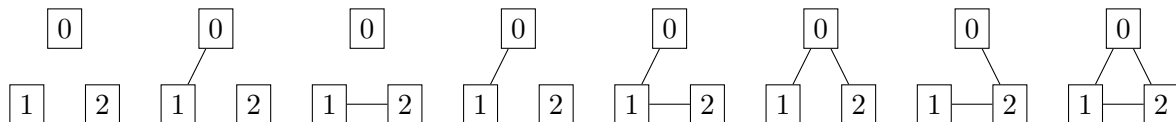
```

## 2. Dictionnaire d'adjacence

```

1 g = {"A": {"B", "C", "D"},
2      "B": {"A", "D"},
3      "C": {"A", "E"},
4      "D": {"A", "B", "E"},
5      "E": {"C", "D"},
6      "F": set()} # par défaut {} représente un dictionnaire

```

**Exercice 2 :****Exercice 3 :**

```

1 def afficher(self)->str:
2     chaine = ""
3     for sommet, voisins in self.sommets.items():
4         chaine += sommet + " { "
5         for voisin in voisins:
6             chaine += voisin + " "
7         chaine += "}\n"
8     return chaine
9
10 # seconde possibilité
11 def __str__(self):
12     chaine = ""
13     for s in self.sommets:
14         if len(self.sommets[s]) == 0:
15             # pour afficher {} au lieu de set()
16             chaine += s + " " + "{}"
17         else:
18             chaine += s + " " + str(self.sommets[s]) + "\n"
19     return chaine

```

**Exercice 4 :**

```
1 def nb_sommets(self)->int:
2     return len(self.sommets)
```

**Exercice 5 :**

```
1 def degre(self, sommet: str)->int:
2     return len(self.sommets[sommet])
```

**Exercice 6 :**

1.  $\sum \text{degrés} = 2 \times \sum \text{arêtes}$  On remarque également que la somme des degrés est nécessairement paire.
2. Cette propriété nous est utile dans l'écriture de la méthode.

```
1 def nb_aretes(self)->int:
2     nb = 0
3     for s in self.sommets:
4         nb += self.degre(s)
5     return nb//2
```

**Exercice 7 :**

```
1 def supprimer(self, sommet: str)->None:
2     for voisin in self.sommets[sommet]:
3         # on supprime sommet dans tous ses voisins
4         self.sommets[voisin].remove(sommet)
5     # on supprime sommet
6     del self.sommets[sommet]
```