

```
1 class Graphe:
2     """
3     Crée un graphe sous forme de dictionnaire d'adjacence
4     """
5
6     def __init__(self):
7         self.sommets = {}
8
9     def ajouter_sommet(self, s)->None:
10        if not(s in self.sommets):
11            self.sommets[s] = set()
12
13    def ajouter_arete(self, s1, s2)->None:
14        self.ajouter_sommet(s1)
15        self.ajouter_sommet(s2)
16        self.sommets[s1].add(s2)
17        self.sommets[s2].add(s1)
18
19    def sont_relies(self, s1, s2)->bool:
20        return s1 in self.sommets[s2]
21
22    def get_adjacents(self, s)->set:
23        return self.sommets[s]
24
25    def get_sommets(self)->list:
26        return list(self.sommets)
```

```
1 from mod_graphe import Graphe
2 import json
3
4 parcours_noir = Graphe()
5
6 with open("parcours_noir.json", "r") as fichier:
7     data = json.load(fichier)
8     """
9     data["sommets"] est une liste de dictionnaires
10    chacun de ces dictionnaires contient:
11        le sommet
12        la liste des ses adjacents
13    """
14    for dico in data["sommets"]:
15        for adj in dico["adjacents"]:
16            parcours_noir.ajouter_arete(dico["sommet"], adj)
17
18 print(parcours_noir.get_sommets())
19 print(parcours_noir.get_adjacents(10))
```