

# TD 10 - Les graphes - correction

## I - Utilisation de la bibliothèque NetworkX

---

1. -
2. -
3. -
4. -
5. Voici les réponses:

```

1  >>> graphe1.is_directed()
2  indique True ou False selon si le graphe est orienté ou non
3  >>> graphe1.degree('a')
4  indique le degré du sommet 'a'
5  >>> graphe1.degree()
6  indique le degré de chaque sommet sous forme d'une collection
7  >>> print(nx.info(graphe1))
8  donne les informations du graphe
9  >>> print(nx.info(graphe1,'a'))
10 donne les informations du sommet 'a'
11 >>> nx.is_empty(graphe1)
12 indique True ou False selon si le graphe est vide ou non
13 >>> graphe1.nodes()
14 indique l'ensemble des étiquettes des sommets sous forme d'une collection
15 >>> graphe1.number_of_nodes()
16 indique le nombre de sommet du graphe
17 >>> graphe1.neighbors('a')
18 indique les voisins du sommet 'a' sous forme d'un objet de type dict_keyiterator
19 >>> list(graphe1.neighbors('a'))
20 indique la liste des voisins (ou successeurs) du sommet 'a'
21 >>> list(nx.all_neighbors(g1,'a'))
22 indique la liste de tous voisins (ou successeurs et prédécesseurs) du sommet 'a'
23 >>> list(nx.non_neighbors(g1,'a'))
24 indique la liste des sommets qui ne sont pas voisins (ou successeurs) du sommet 'a'
25 >>> list(nx.common_neighbors(g1,'a','c'))
26 indique la liste des voisins communs des sommets 'a' et 'c'. Ne fonctionne pas sur
27 >>> graphe1.edges()
28 donne une collection des arêtes (ou arcs) du graphe
29 >>> graphe1.number_of_edges()
30 donne le nombre d'arêtes (ou arcs) dans le graphe
31 >>> list(nx.non_edges(g1))
32 indique la liste des couples qui ne sont pas des arêtes (ou arcs)
33 >>> list(nx.all_simple_paths(g1,'a','e'))
34 donne la liste de toutes (tous) les chaînes (chemins) simple menant de 'a' à 'e'
35 >>> list(nx.simple_cycles(g1))
36 donne la liste de tous les cycles (circuits) du graphe

```

## II - Les réseaux sociaux

---

Voir le fichier `reseaux_sociaux.py` .

## III - les cartes routières

---

Voir le fichier `plus_court_chemin.py` .

