EXERCICE 1 (6 points)

Cet exercice porte sur la programmation objet en Python et les graphes.

Nous avons représenté sous la forme d'un graphe les liens entre cinq différents sites Web :

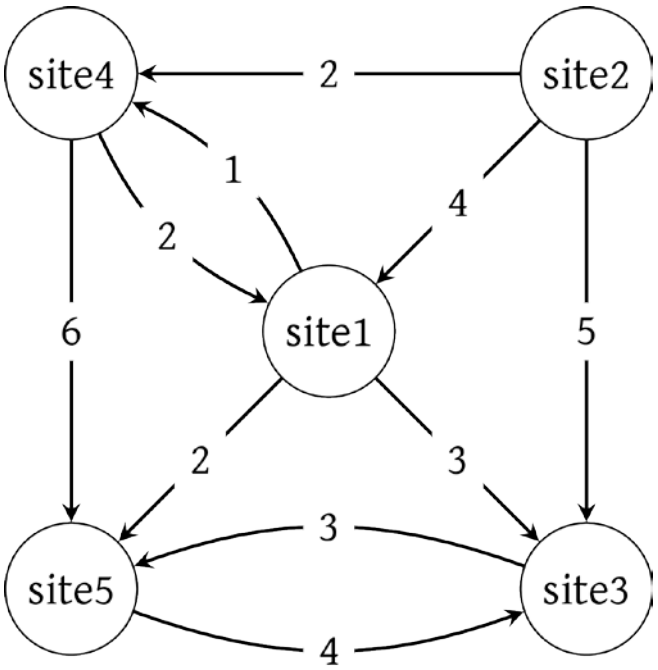

Figure 1. Graphe avec 5 sites

La valeur de chaque arête représente le nombre de citations (de liens hypertextes) d'un site vers un autre. Ainsi, le site **site4** contient 6 liens hypertextes qui renvoient vers le site **site5**.

Les sites sont représentés par des objets de la classe `Site` dont le code est partiellement donné ci-dessous. La complétion de la méthode `calculPopularite` fera l'objet d'une question ultérieure

```
1  class Site:
2
3      def __init__(self, nom):
4          self.nom = nom
5          self.predecesseurs = []
6          self.successeurs = []
7          self.popularite = 0
8          self.couleur = 'blanche'
9
10     def calculPopularite(self):
11         ...
```

Le graphe précédent peut alors être représenté ainsi :

```
1   # Description du graphe
2   s1, s2, s3, s4, s5 = Site('site1'), Site('site2'),
Site('site3'), Site('site4'), Site('site5')
3   s1.successeurs   = [(s3,3), (s4,1), (s5,3)]
4   s2.successeurs   = [(s1,4), (s3,5), (s4,2)]
5   s3.successeurs   = [(s5, 3)]
6   s4.successeurs   = [(s1,2), (s5,6)]
7   s5.successeurs   = [(s3,4)]
8   s1.predecesseurs = [(s2,4), (s4,2)]
9   s2.predecesseurs = []
10  s3.predecesseurs = [(s1,3), (s2,5), (s5,4)]
11  s4.predecesseurs = ...
12  s5.predecesseurs = ...
```

1. Expliquer la ligne 9 de ce code.

2. Les lignes 11 et 12 de cette description du graphe ne sont pas complètes. Recopier et compléter le code des lignes 11 et 12.

3. Donner et expliquer la valeur de l'expression suivante :

```
s2.successeurs[1][1]
```

Pour mesurer la pertinence d'un site, on commence par lui attribuer un nombre appelé valeur de popularité qui correspond au nombre de fois qu'il est cité dans les autres sites, c'est-à-dire le nombre de liens hypertextes qui renvoient sur lui. Par exemple, la valeur de popularité du site **site4** est 3.

4. Donner, selon cette définition, la valeur de popularité du site **site1**.

5. Écrire sur votre copie le code de la méthode `calculPopularite` de la classe `Site` qui affecte à l'attribut `popularite` la valeur de popularité correspondante et renvoie cet attribut.

Afin de calculer cette valeur de popularité pour chacun des sites, nous allons faire un parcours dans le graphe de façon à exécuter la méthode `calculPopularite` pour chacun des objets.

Voici le code de la fonction qui permet le parcours du graphe :

```
1  def parcoursGraphe(sommetDepart):
2      parcours = []
3      sommetDepart.couleur = 'noire'
4      listeS = []
5      listeS.append(sommetDepart)
6      while len(listeS) != 0:
7          site = listeS.pop(0)
8          site.calculPopularite()
9          parcours.append(site)
10         for successeur in site.successeurs:
11             if successeur[0].couleur == 'blanche':
12                 successeur[0].couleur = 'noire'
13                 listeS.append( successeur[0] )
14     return parcours
```

On rappelle les points suivants :

- la méthode `append` ajoute un élément à une liste Python ;

  par exemple, `tab.append(el)` permet d'ajouter l'élément `el` à la liste Python `tab` ;

- la méthode `pop` enlève de la liste l'élément situé à la position indiquée et le renvoie en valeur de retour;

  par exemple, `tab.pop(2)` enlève l'élément à l'indice 2 et le renvoie.

Dans ce parcours, les sites non encore traités sont de couleur `'blanche'` (valeur par défaut à la création de l'objet) et ceux qui sont traités de couleur `'noire'`.

6. Dans ce parcours, on manipule la liste Python nommée `listeS` uniquement à l'aide d'appels de la forme `listeS.append(sommet)` et `listeS.pop(0)`. Donner la structure de données correspondant à ces manipulations.

7. Donner le nom de ce parcours de graphe.

8. La fonction `parcoursGraphe` renvoie une liste `parcours`. Indiquer la valeur renvoyée par l'appel de fonction :

   ```
   parcoursGraphe(s1)
   ```

On cherche maintenant le site le plus populaire, celui dont la valeur de popularité est la plus grande.

Voici le code de la fonction qui renvoie le site le plus populaire, elle prend comme argument une liste non vide contenant des instances de la classe `Site`.

```
1  def lePlusPopulaire(listeSites):
2      maxPopularite = 0
3      siteLePlusPopulaire=listeSites[0]
4      for site in listeSites:
5          if site.popularite > maxPopularite:
6              ...
7              ...
8      return siteLePlusPopulaire
```

9. Copier et compléter les lignes 6 et 7 de cette fonction.

10. Donner ce que renvoie la ligne de code suivante :

```
lePlusPopulaire(parcoursGraphe(s1)).nom
```

11. On envisage d'utiliser l'ensemble des fonctions proposées ci-dessus pour rechercher le site le plus populaire parmi un très grand nombre de sites (quelques milliers de sites). Expliquer si ce code est adapté à une telle quantité de sites à traiter. Justifier votre réponse.