

## Implantation d'algorithmes de parcours en Python

---

L'objectif de ce TD est d'obtenir **rapidement** une implantation des algorithmes de parcours de graphe en Python. Le langage est particulièrement approprié pour ce type d'exercice car très proche du « pseudo-code » algorithmique habituel.

### 1 Structures de base

Pour définir une variable de type graphe en Python, on peut utiliser la syntaxe suivante, à base de dictionnaires et de listes, mais surtout très intuitive.

```
graph = {'A': ['B', 'C'],
        'B': ['D'],
        'C': ['D', 'E'],
        'D': [],
        'E': ['F'],
        'F': ['C']}
```

Une fonction qui parcourt les arêtes d'un graphe d'écrit simplement :

```
def mafonction(gr) :
    for x in gr :
        for y in gr[x] :
            ...
```

Afin de manipuler un tableau d'états (ou autre chose), on peut définir une variable globale et s'en servir plus tard dans une fonction :

```
Etats = { }
...
Etats[u] = "NonAtteint"
```

### 2 Parcours en profondeur et en largeur

1. Ecrire une fonction de parcours en profondeur et la tester.
2. Modifier ce parcours pour qu'il simule le déplacement d'un robot, en affichant chaque étape où le robot atteint pour la première fois un sommet, et chaque étape où le robot revient sur un sommet.
3. Ecrire et tester une fonction de parcours en largeur.