

# Structures de Données TP4

B. Pulfer, A. Freeman, H. Haldi<sup>1</sup>

<sup>1</sup>Département d'Informatique UNIGE

Mai 2023

## But

Le but de ce TP est de manipuler graphes et les algorithmes principaux de recherche dans ces structures. Pour les schémas, diagrammes d'activité et fiches demandées, utilisez [nomnoml](#) ou [draw.io](#).

Quand il s'agit d'implémenter, veuillez vous référer aux diapositives sur le C.

## 1 Notions théoriques

Soit un graphe **orienté**  $G = (V, E)$  avec  $V$  l'ensemble des nœuds du graphe, et  $E = \{(i, j) \mid i, j \in \mathbb{N}\}$ , l'ensemble des transitions d'un nœud  $i$  à un nœud  $j$ .

Notez que  $(i, j)$  veut dire que le nœud  $i$  est relié au nœud  $j$  dans cette direction.



### 1.1 Graphe non-pondéré

Soit  $G = (V, E)$  un graphe orienté avec  $V = \{0, 1, 2, 3, 4, 5, 6\}$  l'ensemble des nœuds et  $E = \{(0, 1), (0, 2), (1, 4), (1, 3), (2, 3), (2, 5), (3, 6), (4, 6), (5, 6)\}$  l'ensemble des arêtes.

1. Dessinez le graphe correspondant.
2. Écrivez le graphe sous forme de matrice d'adjacence et de liste d'adjacence.
3. Expliquez le fonctionnement général, les différences et l'utilité des procédures DFS et BFS.
4. Donnez un exemple de l'application de chacune de ces procédures sur le graphe donné. Assurez vous de bien fournir l'ordre de parcours du graphe pour chacune des procédures.  
( $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots$ )
5. Donnez un scénario réel où un BFS est préférable à un DFS.

### 1.2 Graphe pondéré

Pour chacune des arêtes du graphe une pondération  $C(i, j)$  représentant la distance où le coût entre le nœud  $i$  et  $j$ , est définie comme suit,

$(i, j)$	(0,1)	(0,2)	(1,4)	(1,3)	(2,3)	(2,5)	(3,6)	(4,6)	(5,6)
$C(i, j)$	0	1	5	3	2	1	1	7	3

1. Définissez la matrice d'adjacence représentant le graphe en prenant en compte ces distances.
2. Donnez un exemple du monde réel de ce que cette matrice et ce graphe pourraient représenter.
3. Expliquez le principe de fonctionnement et l'utilité d'un algorithme du plus court chemin.
4. Donnez un exemple détaillé de l'application d'un tel algorithme sur le graphe donné.

## 2 Modélisation et Implémentation

Nous voulons maintenant reproduire ce que nous venons de faire sur papier dans un programme (pas de squelette cette fois ☺).

1. Modélisez un graphe via une **liste d'adjacence** à l'aide des TADs. Explicitez les opérations que vous allez implémenter sur les graphes (e.g. rajout de noeud, rajout d'arête, suppression...)
2. Implémentez les procédures BFS et DFS à l'aide de votre **modélisation**.
3. **Bonus:** Attribué à l'implémentation qui peut exécuter BFS et DFS sur n'importe quelle matrice / liste d'adjacence carrée.