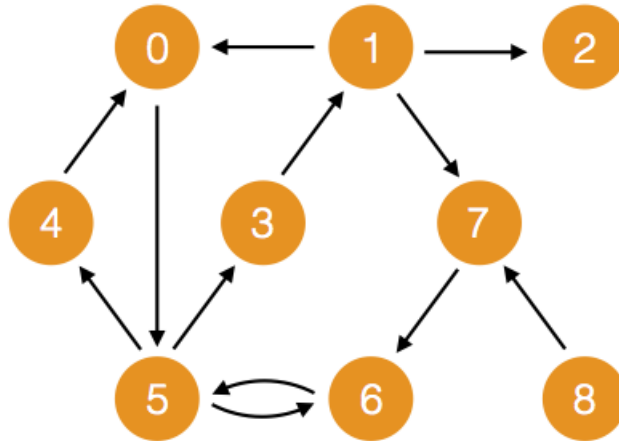


# Algorithmique des graphes TD1

## 1. Algorithmes de parcours



**1.1** Effectuez un parcours en profondeur récursif en indiquant la nature de chaque arc (liaison, avant, transverse ou arrière). Indiquez les dates de début et fin.

**1.2** Effectuez un parcours en profondeur itératif avec pile. Indiquez les dates de début et l'état de la pile, comme vu en cours.

**1.3** Effectuez un parcours en largeur avec file d'attente. Indiquez les dates de début et l'état de la file, comme vu en cours.

## 2. Tri topologique

**2.1** Appliquez l'algorithme du tri topologique par parcours en profondeur aux graphes décrits par les figures 1 et 2.

**2.2** Proposez une extension de cet algorithme permettant d'afficher un cycle si le graphe en contient et que le tri topologique est alors impossible à réaliser. Expliquez.

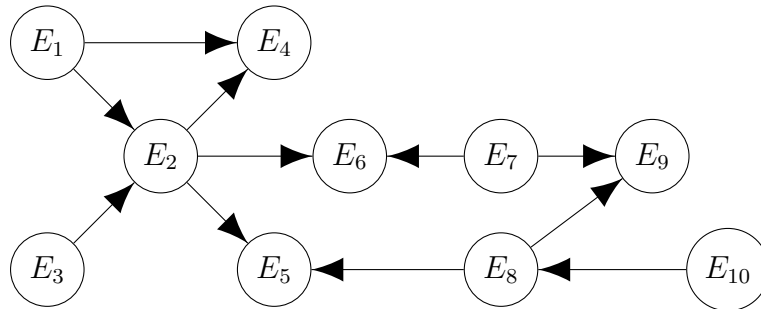


FIGURE 1 – Graphe 1

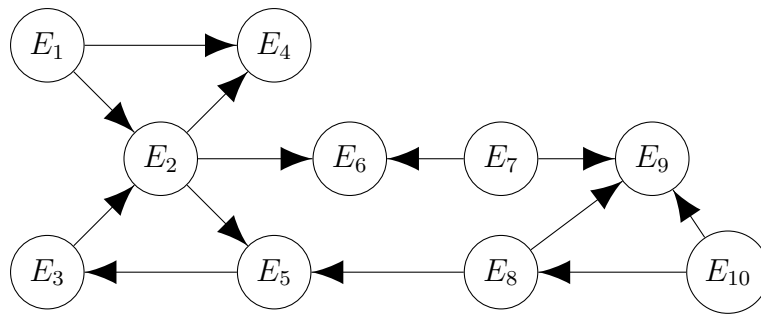


FIGURE 2 – Graphe 2

### 3. Tri topologique appliqué à l'ordonnancement

On considère un programme constitué de 12 modules. L'exécution du programme impose le séquençement décrit dans le tableau 1 :

$x$	1	2	3	4	5	6	7	8	9	10	11	12
$preds(x)$	10, 11	4, 9	9	1	1, 9	2, 5	2	2		12	12	

**3.1** Appliquez la version "points d'entrée" de l'algorithme de Kahn de tri topologique à ce graphe.

**3.2** Appliquez la version "points de sortie" de l'algorithme de Kahn de tri topologique à ce graphe.

**3.3** Si l'exécution de chaque module dure 1ms, en combien de temps peut s'exécuter le programme ?