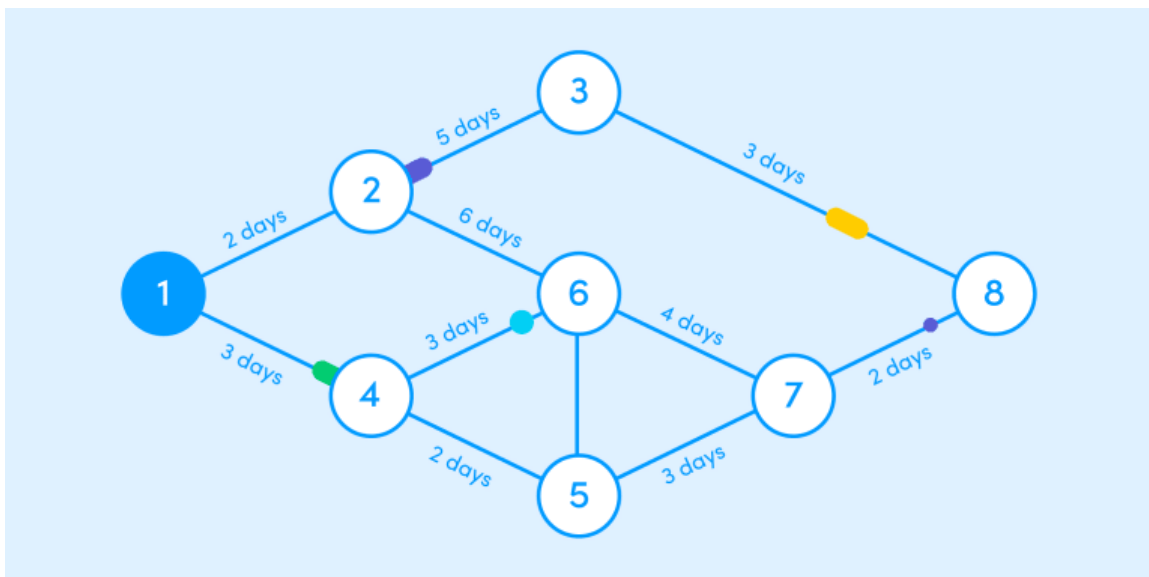


# LA PRÉPA DES INP

RAPPORT DE PROJET : MINEURE INFORMATIQUE

## **PERT** Program Evaluation and Review Technique

### Gestion d'un projet par un PERT



FERRER Paul  
EL KOURI Yassine  
BIKRIA Badr

*Sous la supervision de*  
M. CREGUT  
M. FILALI

# 1 Problématique

On considère un projet composé de tâches auxquelles on associe une durée d'exécution et des relations avec d'autres tâches. Chaque tâche nécessite que les tâches qui la précèdent aient été exécutées pour pouvoir débiter. Le but du projet est de créer des structures de données et des programmes permettant d'évaluer la faisabilité des projets et les conditions en termes de temporalité.

## 2 Architecture du programme

Notre programme est divisé en deux fichiers. Le fichier *fonctions\_principales.py* définit la classe Graphe et contient toutes les fonctions sauf celle du rendu de l'analyse qui, elle, est contenue dans le fichier *analyse.py*.

### 2.1 Documentation des fonctions

- `contient_cycle` : permet de détecter si un graphe contient des cycles.  
Prend en entrée un graphe et renvoie un booléen.
- `tous_chemins` : renvoie une liste contenant tous les chemins possibles entre deux points. Est nécessaire pour déterminer les chemins critiques.  
Prend en entrée un graphe, les points de départ et d'arrivée et des listes contenant les noeuds visités, le chemin parcouru et l'ensemble des chemins parcourus. Ces trois listes sont nécessaires car la fonction est récursive.
- `Afficher_tous_chemins` : permet de renvoyer le résultat de *tous\_chemins* sans avoir à spécifier les listes vides.  
Prend en entrée un graphe et les points de départ et d'arrivée.
- `lire_taches` : permet de placer les données contenues dans le fichier de tâches dans une liste. Prend en entrée le fichier contenant les tâches et leurs informations et renvoie une liste de tâches. Ces tâches sont représentées par une liste contenant dans l'ordre suivant : identifiant de la tâche, durée de la tâche, description de la tâche.
- `lire_liaisons` : permet de placer les données contenues dans le fichier de liaisons dans un ensemble.  
Prend en entrée le fichier contenant les liaisons et renvoie un ensemble de couples de la forme (*point de départ*, *point d'arrivée*)
- `duree_taches` : crée un dictionnaire associant à chaque tâches sa durée.  
Prend en entrée le fichier de tâches et renvoie le dictionnaire.
- `generer_graphe` : crée le graphe associé au système de tâches contenu dans les fichiers fournis.  
Prend en entrée les fichiers de tâches et de liaisons et renvoie le graphe associé.
- `arc_present` : permet de savoir s'il existe un chemin entre deux points.  
Prend en entrée un graphe et deux points et renvoie un booléen disant si oui ou non il y a un chemin entre ces deux points.
- `parcoursProfondeur` : Renvoie le parcours en profondeur d'un graphe sous forme de liste.
- `chemin_critique` : trouve les chemins critiques, c'est-à-dire les chemins contenant les points critiques (les points dont le retard entraînera un retard global du projet), d'un graphe.  
Prend en entrée un graphe, un point de départ, un point d'arrivée et le fichier de tâches relatif au graphe.  
Renvoie un doublet formé par la liste du/des chemin/s critique/s et de la durée critique.
- `triTopologique` : renvoie le tri topologique des noeuds d'un graphe, c'est-à-dire tous les noeuds du graphe triés du début à la fin, par niveaux.

Prend en entrée le graphe.

- `date_au_plus_tot` : calcule la date de départ possible au plus tôt pour chaque tâches.  
Prend en entrée un graphe et le fichier de tâche associé. Retourne un dictionnaire associant à chaque tâche sa date de commencement au plus tôt.
- `date_au_plus_tard` : calcule la date de départ au plus tard pour chaque tâche sous peine d’engendrer un retard global du projet. Prend en entrée un graphe et le fichier de tâches associé. Retourne un dictionnaire associant à chaque tâche sa date de départ au plus tard.
- `rediger_rapport` : crée un fichier  $\text{\LaTeX}$  contenant les analyses demandées à l’aide des programmes précédents.  
Prend en entrée les fichiers de liaison et de tâche d’un projet.

### 3 Structures de données utilisées

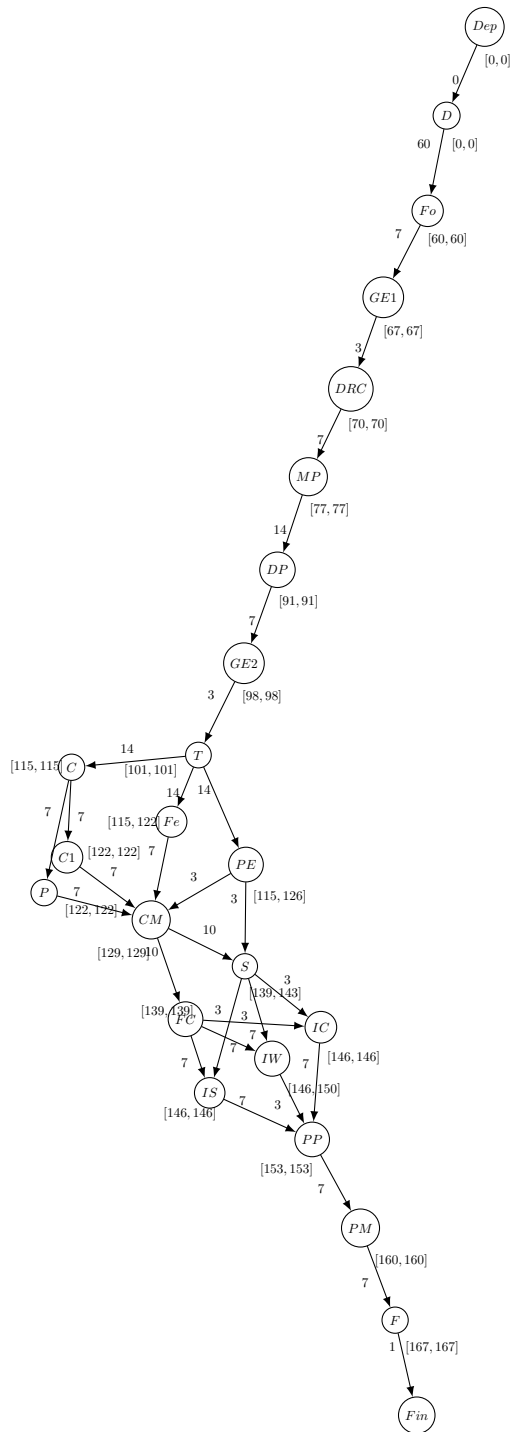
Nous utilisons la classe Graphe vue en cours à laquelle nous avons ajouté des méthodes servant par exemple à détecter la présence d’un cycle dans un graphe. Cette structure est adaptée au problème considéré car on peut représenter une tâche par un noeud du graphe. Ce point sera l’arrivée des arcs issus des tâches précédentes et le départ des arcs allant vers les tâches suivantes.

## 4 Résultats obtenus pour les données de l’annexe A

### 4.1 Faisabilité du projet

Votre projet ne contient pas de cycle, il est faisable.

## 4.2 Visualisation du projet par un graphe



Identifiant	Description
Dep	Depart
D	Permis de construire
Fo	Fondations
GE1	Passage des gaines et evacuations
DRC	Dalle rez de chaussee
MP	Murs porteurs
DP	Dalles plafond
GE2	Passage gaines et evacuation
T	Toiture
Fe	Fenetres
PE	Portes exterieures
IE	Installation electrique et evacuation
C	Chape
C1	Carrelage du sol
P	Parquets
CM	Cloisons et menuiserie interieure
FC	Finition des cloisons
IC	Implantation de la cuisine
IW	Implantation des wc
IS	Implantation des salles de bain
PP	Peinture des plafonds
PM	Peinture des murs
S	Serrurerie
F	Reception de la maison
Fin	Fin

Tâche	Date de début au plus tôt	Date de début au plus tard
Dep	0	0
D	0	0
Fo	60	60
GE1	67	67
DRC	70	70
MP	77	77
DP	91	91
GE2	98	98
T	101	101
PE	115	126
Fe	115	122
C	115	115
P	122	122
C1	122	122
CM	129	129
S	139	143
FC	139	139
IC	146	146
IS	146	146
IW	146	150
PP	153	153
PM	160	160
F	167	167
Fin	168	168

### 4.3 Chemins critiques du projet

Les chemins critiques de votre projet sont les suivants :

['Dep', 'D', 'Fo', 'GE1', 'DRC', 'MP', 'DP', 'GE2', 'T', 'C', 'C1', 'CM', 'FC', 'IS', 'PP', 'PM', 'F', 'Fin']

['Dep', 'D', 'Fo', 'GE1', 'DRC', 'MP', 'DP', 'GE2', 'T', 'C', 'C1', 'CM', 'FC', 'IC', 'PP', 'PM', 'F', 'Fin']

['Dep', 'D', 'Fo', 'GE1', 'DRC', 'MP', 'DP', 'GE2', 'T', 'C', 'P', 'CM', 'FC', 'IS', 'PP', 'PM', 'F', 'Fin']

['Dep', 'D', 'Fo', 'GE1', 'DRC', 'MP', 'DP', 'GE2', 'T', 'C', 'P', 'CM', 'FC', 'IC', 'PP', 'PM', 'F', 'Fin']

Leur duree est de : 168 jours

## 5 Conclusion

Dans l'ensemble, nous avons trouvé ce projet très intéressant. Il nous a permis d'être beaucoup plus à l'aise avec l'utilisation des classes et des dictionnaires. De plus, il nous a permis de découvrir  $\LaTeX$  que ce soit par l'écriture de programmes écrivant des fichiers .tex ou par la rédaction du rapport et du manuel utilisateur. Le projet a été plaisant à faire car nous n'avons bloqué sur aucune question si ce n'est la gestion des graphes avec Overleaf (nous avons finalement utilisé une autre bibliothèque ne posant pas des problèmes de `-shell-escape`). Nous nous sommes répartis de manière assez équitable le travail grâce aux sites de travail en groupe Overleaf et Replit.