# Mathématiques discrètes 2
## Examen du 02/06/2021

**Consignes :**

— Les seuls documents autorisés sont : les supports de cours, et une feuille A4 manuscrite.

— La calculatrice est autorisée.

— Le vocabulaire et les notations adéquates doivent être utilisés.

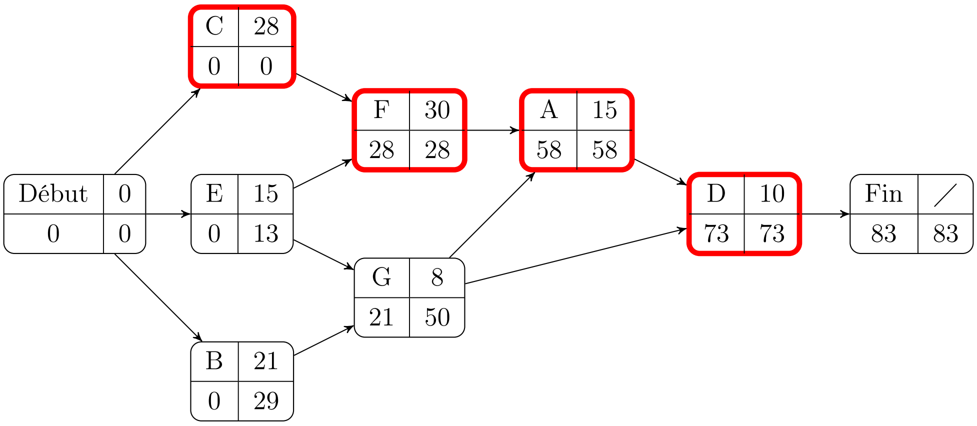
**Exercice 1. Ordonnancement.** (12 points)

Un chantier pour construire une maison passe par différentes phases, découpées en tâches qui sont confiées à des artisans et corps de métier distincts. Afin de planifier l'ensemble de la réalisation, l'ingénieur chargé du chantier a établi le tableau suivant qui liste les différentes tâches par leur code avec leurs contraintes de précédence :

| Code | Durée (jours) | Tâches précédentes |
|---|---|---|
| A | 15 | F,G |
| B | 21 | |
| C | 28 | |
| D | 10 | A,G |
| E | 15 | |
| F | 30 | C,E |
| G | 8 | B,E |

1. À l'aide de la méthode MPM :
   (a) (2 points) représenter le graphe associé au projet (minimiser le nombre de niveaux) ;
   (b) (2 points) donner pour chaque tâche sa date au plus tôt ainsi que sa date au plus tard ;
   (c) (1 point) indiquer les tâches critiques et donner (au moins) un chemin critique.

**Solution:**




Chemin critique : $(C, F, A, D)$.

2. (0,5 points) Existe-t-il des contraintes inutiles dans le projet ? Si oui lesquelles et pourquoi ?

**Solution:** La contrainte $(G, D)$ est inutile car il existe déjà un chemin de $G$ vers $D$ en passant par $A$. Il n'y en a pas d'autres.

3. (0,5 points) Donner la durée minimale totale du projet.

**Solution:** La durée minimale du projet est de 83 jours.

4. (a) (3 points) Calculer les marges totales, libres et certaines de chacune des tâches du projet.
*Rappel : la marge certaine d'une tâche $k$ est donnée par*

$$\mathrm{MC}(k) = \begin{cases} \min\limits_{j \in \Gamma^+(k)} t_j - \left( \max\limits_{i \in \Gamma^-(k)} T_i + d_i \right) - d_k & \textit{si positif} \\ 0 & \textit{sinon} \end{cases}$$

**Solution:**

| Tâches | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| Marge totale | 0 | 29 | 0 | 0 | 13 | 0 | 29 |
| Marge libre | 0 | 0 | 0 | 0 | 6 | 0 | 29 |
| Marge certaine | 0 | 0 | 0 | 0 | 6 | 0 | 0 |

Répondre aux questions suivantes en justifiant à l'aide des marges précédemment calculées.

(b) (0,5 points) Existe-t-il une tâche dont la date de début peut être décalée sans décaler la fin du projet ? Pourquoi ?

**Solution:** Toute tâche $k$ telle que $\mathrm{MT}(k) > 0$ peut être décalée dans décaler la fin du projet, par exemple ici les tâches B, E et G.

(c) (0,5 points) Existe-t-il une tâche dont la date de début peut être décalée sans impacter aucune des tâches suivantes ? Pourquoi ?

**Solution:** Toute tâche $k$ telle que $\mathrm{ML}(k) > 0$ peut être décalée dans décaler la date de début d'aucune des tâches suivantes, par exemple ici les tâches E et G.

(d) (1 point) Existe-t-il une tâche dont la date de début peut être décalée sans impacter aucune des tâches suivantes même si des décalages dans les tâches qui la précédent ont déjà eu lieu ? Pourquoi ?

**Solution:** Toute tâche $k$ telle que $\mathrm{MC}(k) > 0$ peut être décalée dans décaler la date de début d'aucune des tâches suivantes, et ceci même si une des tâches qui précédent a déjà subi un décalage, par exemple ici la tâche E.

5. (1 point) Lors de l'implémentation de la méthode MPM, les tâches doivent d'abord être organisées par niveaux. Est-il préférable de minimiser le nombre de niveaux nécessaires, ou bien de maximiser le nombre de niveaux ? Pourquoi ?

**Solution:** Si $n$ est le nombre de tâches d'un projet, et $m$ le nombre de contraintes de précédence (i.e. d'arcs).

Alors minimiser le nombre de niveaux coûte un temps $O((n+m)\log n)$ en utilisant la structure de données *tas minimier*.

En revanche un simple parcours en profondeur afin d'effectuer un tri topologique du graphe place chaque sommet dans un niveau distinct et ainsi maximise le nombre de niveaux, cela pour un coût de $O(n+m)$. Il est donc préférable de maximiser le nombre de niveaux pour accélérer le pré-traitement du graphe associé au projet.

**Exercice 2.** (3 points)

Montrer que dans tout projet, et pour toute tâche $k$ critique et différente de Début et Fin, on a

$$\mathrm{MC}(k) = 0$$

avec la définition suivante

$$\mathrm{MC}(k) = \begin{cases} \min\limits_{j\in\Gamma^+(k)} t_j - \left(\max\limits_{i\in\Gamma^-(k)} T_i + d_i\right) - d_k & \text{si positif} \\ 0 & \text{sinon} \end{cases}$$

---

**Solution:** Soit $k$ une tâche critique, i.e. $T_k = t_k$.

L'objectif est de montrer que

$$\begin{aligned} & \min_{j\in\Gamma^+(k)} t_j - \left(\max_{i\in\Gamma^-(k)} T_i + d_i\right) - d_k \leq 0 \\ \Leftrightarrow\quad & \left(\min_{j\in\Gamma^+(k)} t_j\right) - \left(\max_{i\in\Gamma^-(k)} T_i + d_i\right) - d_k \leq 0 \\ \Leftrightarrow\quad & \left(\min_{j\in\Gamma^+(k)} t_j\right) \leq \left(\max_{i\in\Gamma^-(k)} T_i + d_i\right) + d_k \\ \Leftrightarrow\quad & \exists j \in \Gamma^+(k) \quad t_j \leq \left(\max_{i\in\Gamma^-(k)} T_i + d_i\right) + d_k \\ \Leftrightarrow\quad & \exists j \in \Gamma^+(k) \quad t_j - d_k \leq \left(\max_{i\in\Gamma^-(k)} T_i + d_i\right) \\ \Leftrightarrow\quad & \exists j \in \Gamma^+(k)\ \exists i \in \Gamma^-(k) \quad t_j - d_k \leq T_i + d_i \end{aligned}$$

Or pour $j \in \Gamma^+(k)$ telle que $T_k = T_j - d_k$ (existe par défintion de $T_k$) et $i \in \Gamma^-(k)$ telle que $t_k = t_i + d_i$ (existe par définition de $t_k$) on a

$$t_j - d_k \leq T_j - d_k = T_k = t_k = t_i + d_i \leq T_i + d_i$$

Par conséquent on en déduit que $\mathrm{MC}(k) \leq 0$ et donc $\mathrm{MC}(k) = 0$.

---

**Exercice 3. Matrice d'adjacence et algorithme.** (6 points)

Soit $G = (V, E)$ un graphe orienté avec $V = \{v_1, \ldots, v_n\}$, et $M$ sa matrice d'adjacence associée de dimensions $n \times n$. On notera $(m_{i,j})$ les coefficients de la matrice $M$, et $(m^{[p]}_{i,j})$ les coefficients de la matrice $M^{[p]}$.

Le but de l'exercice est d'utiliser les puissances booléennes de la matrice $M$ afin de donner une méthode permettant de calculer la longueur du plus long chemin (en nombre d'arêtes/arcs) entre deux sommets donnés $v_i$ et $v_j$.

Soient $v_i$ et $v_j$ deux sommets de $G$ (non nécessairement distincts). Montrer les affirmations suivantes :

1. (1 point) Il existe un chemin de $v_i$ à $v_j$ *si et seulement si* il existe un entier $p$ dans $[\![1, n-1]\!]$ tel que $m^{[p]}_{i,j} = 1$.

---

**Solution:** Si un chemin de $v_i$ à $v_j$ existe alors on peut en extraire un chemin élémentaire (donc sans circuit) d'une longueur au plus $n-1$ (i.e. qui ne passe que par des sommets tous différents), et donc on aura un $p$ dans $[\![1, n-1]\!]$ tel que $m^{[p]}_{i,j} = 1$.

Il a été vu en TD que $m^{[p]}_{i,j} = 1$ si et seulement si il existe un chemin de longeur $p$ de $v_i$ à $v_j$, donc si $m^{[p]}_{i,j} = 1$ alors il existe un chemin de $v_i$ à $v_j$.

---

2. (1 point) Il existe un chemin possédant un circuit de $v_i$ à $v_j$ *si et seulement si* il existe un entier $p$ dans $[\![n, 2n-1]\!]$ tel que $m_{i,j}^{[p]} = 1$.

> **Solution:** Supposons qu'il existe un chemin possédant un circuit de $v_i$ à $v_j$. On peut en extraire un chemin élémentaire $c_1$ de $v_i$ à $v_j$ de longueur inférieure strictement à $n-1$ et un circuit élémentaire $c_2$ d'un sommet $v_k$ de $c_1$ vers lui-même et donc de longueur inférieure ou égale à $n$. On peut alors construire un chemin de $v_i$ à $v_j$ en suivant les sommets de $c_1$ puis, arrivé sur $v_k$, répéter autant de fois que nécessaire le circuit $c_2$, et enfin terminer le chemin $c_1$ afin d'obtenir un chemin de $v_i$ à $v_j$ de longueur comprise entre $n$ et $2n-1$ (par le principe des tiroirs ou du théorème des valeurs intermédiaires discret).
>
> S'il existe un entier $p \in [\![n, 2n-1]\!]$ tel que $m_{i,j}^{[p]} = 1$ alors il existe un chemin de longueur $p \geq n$ de $v_i$ à $v_j$ passant donc plusieurs fois par le même sommet et donc possédent un circuit.

3. (1 point) En déduire une méthode qui, étant donné une matrice d'adjacence d'un graphe orienté et deux sommets $v_i$ et $v_j$, calcule la longueur du plus long chemin de $v_i$ à $v_j$ : si aucun chemin n'existe le résultat doit être 0 et si un chemin possédant un circuit existe le résultat doit être $\infty$.
*Il n'est pas nécessaire d'écrire la méthode en pseudo-code.*

> **Solution:** La méthode consiste alors à calculer successivement les puissantes de $M^{[p]}$ pour $p$ de 1 à $2n-1$ et à mémoriser la puissance de $p$ la plus grande telle que $m_{i,j}^{[p]} = 1$, qu'on note $p^\star$ dans la suite. On a alors trois cas :
>
> — si $p^\star = 0$, i.e. aucune des puissances de 1 à $n-1$ ne contient un 1 en ligne $i$ alors il n'y a pas de chemin de $v_i$ à $v_j$ et le résultat est 0 ;
>
> — si $0 < p^\star < n$, i.e. il n'y a pas de circuit sur les chemins de $v_i$ à $v_j$ donc le résultat est $p^\star$ ;
>
> — si $n \leq p^\star$, i.e. il y a un circuit sur un des chemins de $v_i$ à $v_j$ donc le résultat est $\infty$.

4. (1 point) Analyser le temps de calcul nécessaire en fonction du nombre de sommets $n$ du graphe $G$.

> **Solution:** Il y a $2n-2$ multiplications de matrices $n \times n$ à effectuer, chacune en temps $O(n^3)$ plus un traitement en temps constant (vérification d'une case et mise à jour de $p^\star$). Au total la méthode prend donc un temps de $O(n^4)$.

5. (1 point) La méthode proposée s'adapte-t-elle si le graphe $G$ est non-orienté ? Si oui le temps de calcul est-il modifié ?

> **Solution:** Toute l'analyse précédente reste valide en remplaçant les termes « chemin » et « circuit » respectivement par « chaîne » et « cycle ». La matrice est symétrique mais le temps de calcul d'une multiplication reste asymptotiquement le même, donc le temps de calcul total asymptotique est aussi en $O(n^4)$.

6. (1 point) Est-il possible de trouver une méthode plus efficace en temps de calcul en utilisant des parcours du graphe ? Si oui laquelle et pour quel temps de calcul ?

> **Solution:** Un parcours en largeur permet de trouver les chemins de longueur minimale entre un sommet d'origine et tous les autres, et aussi détecter des circuits sur le chemin de l'origine à un autre sommet.
>
> Un parcours en profondeur permet de tester l'existence d'un chemin entre un sommet d'origine et tous les autres, et aussi détecter des circuits sur le chemin de l'origine à un autre sommet.
>
> Par conséquent il ne paraît pas immédiat d'utiliser ici un simple parcours de graphe.