Contraintes et Planification

Épreuve pratique d'algorithmique et de programmation
Concours commun des Écoles normales supérieures

Durée de l'épreuve: 3 heures 30 minutes

Juin/Juillet 2016

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Il vous a été donné un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion!). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de u_0. Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures!
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple: $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe***. Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

Le sujet est constitué de trois sections. Le début de chaque section est indépendante du reste du sujet.

On considère les suites suivantes. Soit u_0 indiqué sur votre feuille. On définit

$$\begin{aligned}
&\forall t \geq 1, u_t = a \cdot u_{t-1} \bmod b,\\
&\forall t \geq 0, u_{t,0} = u_t,\\
&\forall t \geq 0, i \geq 1, u_{t,i} = c \cdot u_{t,i-1} \bmod d,\\
&\forall t \geq 0, i \geq 0, u_{t,i,0} = u_{t,i},\\
&\forall t, i \geq 0, j \geq 1, u_{t,i,j} = e \cdot u_{t,i,j-1} \bmod f,
\end{aligned}$$

On prendra $a = 263, b = 269, c = 271, d = 281, e = 283, f = 2879$.

Question 1 *Calculer* ***a)*** u_1 ***b)*** u_{50} ***c)*** $u_{50,50}$ ***d)*** $u_{50,1000,500}$.

Pour tout $a, b \in \mathbb{N}$, on note $[a, b] = \{a, a+1, \ldots, b\}$.

1 Contraintes de différence

On considère un ensemble d'inconnues $\{x_0, x_1, \ldots, x_n\}$ qui représentent chacune la date d'un évènement. On suppose que la variable x_0 est toujours égale à zéro. Soit $c_{i,j} \in \mathbb{R} \cup \{\infty\}$ pour tout $0 \leq i, j \leq n$. On considère le système de contraintes $S = \big((x_i)_{i\in[0,n]}, (c_{i,j})_{i,j\in[0,n]}\big)$, qui représente les inégalités suivantes :

$$\begin{aligned}
&x_0 = 0,\\
&\forall i, j \in [0, n], x_i - x_j \leq c_{i,j}.
\end{aligned}$$

On dit qu'une inconnue x_i est bornée si $c_{i,0} < \infty$. Le système de contraintes est dit satisfiable s'il existe une fonction $v \in [0, n] \to \mathbb{R}$ telle que $v(0) = 0$ et $\forall i, j \in [1, n], v(x_i) - v(x_j) \leq c_{i,j}$. Dans ce cas, on appelle v une valuation de S.
Par exemple, le système suivant définit le système de contraintes avec les inconnues (x_1, x_2, x_3), et les poids $c_{1,2} = 1, c_{2,3} = 1$, et $c_{i,j} = \infty$ pour tout autre i, j.

$$x_1 - x_2 \leq 1, x_2 - x_3 \leq 1. \tag{1}$$

Ce système est satisfiable. En effet, une valuation v peut être définie par $v(0) = 0, v(1) = 0, v(2) = 0, v(3) = 0$, mais aussi par $v(0) = 0, v(1) = 1, v(2) = 0, v(3) = -1$.

Le but de cette section est de déterminer si un système de contraintes donné est satisfiable, et d'en calculer une valuation témoin.
On définit le système $\mathcal{S}(n, t)$ avec $t \geq 1$, sur les inconnues $\{x_0, x_1, \ldots, x_n\}$ avec les contraintes définies par

$$c_{i,j} = \begin{cases} 0 & \text{si } i = j,\\ u_{t,i,j} - \lfloor f/2 \rfloor & i \neq 0 \text{ et } u_{t-1,i,j} \equiv 0 \bmod n,\\ u_{t,i,j} - \lfloor f/2 \rfloor & i = 0 \text{ et } (u_{t-1,i,j} \bmod n) \leq n/3,\\ \infty & \text{sinon.} \end{cases}$$

Pour n et t donnés, on note $i_{\min}$ l'indice minimale d'une inconnue bornée de $\mathcal{S}(n, t)$, et $i_{\max}$ l'indice maximale d'une inconnue bornée de $\mathcal{S}(n, t)$. Autrement dit, $i_{\min} = \min\{i \in [1, n] \mid c_{i,0} < \infty\}$, et $i_{\max} = \max\{i \in [1, n] \mid c_{i,0} < \infty\}$. Ces indices valent ∞ si $\mathcal{S}(n, t)$ n'a pas d'inconnue bornée.

Question 2 *Pour n donné, calculer la plus petite valeur de $t \in [1, 20]$ pour laquelle les indices $(i_{\min}, i_{\max})$ associés à $\mathcal{S}(n, t)$ sont finis. Écrire le triplet de valeur $(t, i_{\min}, i_{\max})$ s'il existe une telle valeur de t, et $\emptyset$ sinon.*
a) $n = 50$ ***b)*** $n = 100$ ***c)*** $n = 150$.

1.1 Satisfaction

Pour tout nombre v, soit $\vec{v}$ la valuation constante égale à v, c'est-à-dire : $\forall i \in \{0, 1, \ldots, n\}, \vec{v}(i) = v$.
Étant donné $S = \big((x_i)_{i\in[0,n]}, (c_{i,j})_{i,j\in[0,n]}\big)$, et $0 \leq m \leq n$, on définit le sous-système de S de taille m comme le système de contraintes $S_m = \big((x_i)_{i\in[0,m]}, (c_{i,j})_{i,j\in[0,m]}\big)$, autrement dit, c'est un système obtenu en ne gardant que les inconnues $x_0, \ldots, x_m$.

Question 3 *Pour les systèmes suivants, écrire la taille maximale d'un sous-système dont $\vec{0}$ est une valuation.* ***a)*** $\mathcal{S}(100, 1)$, ***b)*** $\mathcal{S}(100, 2)$, ***c)*** $\mathcal{S}(100, 3)$.

1.2 Satisfiabilité

Étant donné un système $S = \big((x_i)_{i\in[0,n]}, (c_{i,j})_{i,j\in[0,n]}\big)$, on lui associe un graphe orienté pondéré $\mathbf{G}(S) = (V, E, w)$ où les sommets sont $V = \{0, 1, \ldots, n\}$, et les arêtes $E = \{(i, j) \mid 0 \leq i, j \leq n\}$ et les poids sont associés aux arêtes par $w((i, j)) = c_{i,j}$.
Un cycle $s_1 s_2 \ldots s_k s_{k+1}$ de G, avec $s_{k+1} = s_1$, est dit négatif si la somme des poids de ses arêtes est strictement négative : $\sum_{i=1}^{k} w(s_i, s_{i+1}) < 0$.

Question à développer pendant l'oral 1 *Démontrer que si le graphe contient un cycle négatif, alors le système de contraintes n'est pas satisfiable.*

On admet le théorème suivant.

Théorème 1 *Un système de contraintes S est satisfiable si et seulement si $\mathbf{G}(S)$ ne contient pas de cycle négatif.*

La satisfiabilité d'un système de contraintes revient donc à détecter l'existence d'un cycle négatif dans un graphe. Nous allons utiliser l'algorithme de Floyd-Warshall pour cette tâche.
On rappelle que sur les graphes sans cycles négatifs, l'algorithme de Floyd-Warshall calcule la longueur du plus court chemin entre toutes les paires de sommets. Plus précisément, il calcule un tableau `c` où `c`$[i][j]$ est la longueur du plus court chemin de i vers j, pour tous sommets i, j. On rappelle l'algorithme de Floyd-Warshall :

```
Entree : une matrice d'adjacence c[0..n][0..n]

for k = 0 to n do
  for i = 0 to n do
    for j = 0 to n do
      if(c[i][j] > c[i][k] + c[k][j])
         c[i][j] <- c[i][k] + c[k][j]
```

Pour un système de contraintes S, soit $\mathtt{c}_S$ le tableau calculé par cet algorithme sur le graphe $\mathbf{G}(S)$. On notera que $\mathtt{c}_S[i][j]$ n'est pas nécessairement la longueur du plus court chemin de i à j puisque $\mathbf{G}(S)$ pourrait contenir des cycles négatifs. On a cependant la propriété suivante :

Théorème 2 *Soit $\mathtt{c}_S$ le tableau calculé par l'algorithme de Floyd-Warshall pour $\mathbf{G}(S)$. Alors, $\mathbf{G}(S)$ contient un cycle négatif si et seulement s'il existe $i \in \{0, \ldots, n\}$ tel que $\mathtt{c}_S[i][i] < 0$.*

Question 4 *Écrire un algorithme qui détermine si un système de contraintes donné est satisfiable. Pour les systèmes suivants, écrire `vrai` s'il est satisfiable, `faux` sinon.* ***a)*** $\mathcal{S}(200, 1)$ ***b)*** $\mathcal{S}(300, 2)$ ***c)*** $\mathcal{S}(400, 3)$.

1.3 Forme Canonique

On peut remarquer qu'un même ensemble de solutions est représenté par plusieurs systèmes de contraintes différentes. Par exemple, le système

$$x_1 - x_2 \leq 1, x_2 - x_3 \leq 1, x_1 - x_3 \leq \infty \tag{2}$$

représente les mêmes solutions que

$$x_1 - x_2 \leq 1, x_2 - x_3 \leq 1, x_1 - x_3 \leq 2. \tag{3}$$

En effet, les deux systèmes sont "équivalents" puisque $x_1 - x_3 \leq 2$ peut être obtenu en sommant les deux premières inégalités.
On définit la forme canonique d'un système de contrainte $S = \big((x_i)_{i\in[0,n]}, (c_{i,j})_{i,j\in[0,n]}\big)$ comme $S' = \big((x_i)_{i\in[0,n]}, (c'_{i,j})_{i,j\in[0,n]}\big)$ où $c'_{i,j}$ est le poids du plus court chemin de i dans j dans le graphe $\mathbf{G}(S)$.

Question à développer pendant l'oral 2 *Soit un système de contrainte S et sa forme canonique S'. Démontrer que pour tout $v \in [0, n] \to \mathbb{R}$, v est une valuation de S, si et seulement si v est une valuation de S'.*

Étant donné un système $S = \big((x_i)_{i\in[0,n]}, (c_{i,j})_{i,j\in[0,n]}\big)$, soit $S' = \big((x_i)_{i\in[0,n]}, (c'_{i,j})_{i,j\in[0,n]}\big)$ sa forme canonique. On définit l'empreinte de S comme

$$\mathsf{empreinte}(S) = \sum_{i,j\in[0,n]:c'_{i,j}<\infty} |c'_{i,j}| \bmod 101.$$

Question 5 *Pour chaque n et t_0 donnés, calculer le plus petit $t \in [0, 9]$ pour lequel $\mathcal{S}(n, t + t_0)$ est satisfiable. Si un tel t existe, alors écrire la paire $(t, \mathsf{empreinte}(\mathcal{S}(n, t + t_0))$, sinon écrire $\emptyset$.* ***a)*** $n = 10, t_0 = 1$ ***b)*** $n = 50, t_0 = 11$ ***c)*** $n = 100, t_0 = 22$.

Étant donné un système $S = \big((x_i)_{i\in[0,n]}, (c_{i,j})_{i,j\in[0,n]}\big)$ et une contrainte $x_i - x_j \leq c'_{i,j}$, on écrit $S \wedge (x_i - x_j \leq c'_{i,j})$ pour le système obtenu en remplaçant $c_{i,j}$ par $\min(c_{i,j}, c'_{i,j})$ dans S.
On admet la propriété suivante des graphes de contraintes :

Théorème 3 *Soit $S = \big((x_i)_{i\in[0,n]}, (c_{i,j})_{i,j\in[0,n]}\big)$ sous forme canonique. Alors, pour tout $-c_{0,i} \leq v \leq c_{i,0}$, le système $S \wedge (x_i - x_0 \leq v) \wedge (x_0 - x_i \leq -v)$ est satisfiable.*

On notera que même si S est sous forme canonique, $S \wedge (x_i - x_0 \leq v) \wedge (x_0 - x_i \leq -v)$ n'est pas nécessairement sous forme canonique.

On considère un ordre lexicographique sur les valuations d'un système de contraintes. Pour deux valuations $f, g : \{0, 1, \ldots, n\} \to \mathbb{R}$, on dit que $f \leq_{\text{lex}} g$ si le mot $f(0)f(1)f(2)\ldots f(n)$ est plus petit dans l'ordre lexicographique que le mot $g(0)g(1)\ldots g(n)$. Formellement, cette condition s'écrit

$$f = g \text{ ou } \exists i \in \{0, \ldots, n\}, f(i) < g(i) \text{ et } \forall j \in \{0, \ldots, i-1\}, f(j) = g(j).$$

On s'intéresse aux valuations positives, c'est-à-dire aux valuations f telles que $f(i) \geq 0$ pour tout $i \in \{0, \ldots, n\}$. Une valuation est positive minimale si elle est minimale pour l'ordre lexicographique parmi les valuations positives.

Question à développer pendant l'oral 3 *Expliquer comment définir un système de contraintes qui représente l'ensemble des valuations positives de $\mathcal{S}(n, t)$.*

Question 6 *Pour chaque n et t_0 donnés, calculer le plus petit $t \in [0, 9]$ pour lequel $\mathcal{S}(n, t + t_0)$ admet une solution positive minimale. Si un tel t existe, écrire la paire $(t, \sum_{i=0}^{n} f(i))$ où f est la solution positive minimale de $\mathcal{S}(n, t + t_0)$, et écrire $\emptyset$ sinon.*
a) $n = 5, t_0 = 1$ ***b)*** $n = 20, t_0 = 11$ ***c)*** $n = 50, t_0 = 22$ ***d)*** $n = 100, t_0 = 33$

Question à développer pendant l'oral 4 *Expliquer votre algorithme et démontrer sa correction.*

1.4 Application : Ordonnancement optimal

On considère le problème d'ordonnancement suivant. On a m machines qui doivent exécuter chacune n tâches dans l'ordre. Notons $X_{i,j}$ la j-ème tâche de la machine i. Le temps d'exécution de la tâche $X_{i,j}$ est un entier $d_{i,j}$. Une machine ne peut pas exécuter plus d'une tâche à un instant donné. De plus, chaque tâche $X_{i,j}$ avec $i < m$ a une dépendance $X_{\mathcal{D}(i,j)}$ avec $\mathcal{D}(i, j) \in [i+1, m] \times [1, n]$: la tâche $X_{i,j}$ ne peut pas commencer son exécution avant la terminaison de la tâche $X_{\mathcal{D}(i,j)}$. On note que la dépendance d'une tâche $X_{i,j}$ se trouve nécessairement sur une machine d'indice strictement supérieur (d'où la condition $\mathcal{D}(i, j) \in [i + 1, m] \times [1, n]$).
Notre but est d'obtenir une exécution, qui consiste en un choix des dates d'exécution qui satisfait les dépendances et l'ordre d'exécution. Formellement, on cherche une fonction $\nu : [1, m] \times [1, n] \to \mathbb{N}$ telle que

$$\forall i \in [1, m], j \in [2, n], \nu(i, j) \geq \nu(i, j-1) + d_{i,j-1}$$
$$\forall i \in [1, m-1], j \in [1, n], \nu(i, j) \geq \nu(\mathcal{D}(i, j)) + d_{\mathcal{D}(i,j)}.$$

La première condition assure que chaque machine exécute au plus une tâche à la fois ($X_{i,j}$ ne commence qu'après la terminaison de $X_{i,j-1}$), et que les tâches sont exécutées dans l'ordre. La deuxième condition assure que les dépendances sont respectées : $X_{i,j}$ ne commence pas avant la terminaison de sa dépendance $X_{\mathcal{D}(i,j)}$.
Étant donnée une exécution v, on dit qu'une tâche $X_{i,j}$ termine avant la date T si $\nu(i, j) + d_{i,j} \leq T$. On dit que l'exécution termine avant la date T si toutes les tâches terminent avant la date T.

Construction des instances du problème On définit le problème d'ordonnancement $\mathcal{O}(t, m, n)$ avec m machines et n tâches par machines comme suit.

$$\forall i \in [1, m], j \in [1, n], d_{i,j} = u_{t,i,j}$$
$$\forall i \in [1, m-1], j \in [1, n], \mathcal{D}(i, j) = \big(i + 1 + (u_{t+1,i,j} \bmod m - i), 1 + (u_{t+2,i,j} \bmod n)\big)$$

Ainsi, dans l'instance $\mathcal{O}(t, m, n)$, la dépendance de la tâche $X_{i,j}$ est la tâche $1 + (u_{t+2,i,j} \bmod n)$ de la machine $i+1+(u_{t+1,i,j} \bmod m-i)$. On notera que $i+1+(u_{t+1,i,j} \bmod m-i) \in \{i+1, i+2, \ldots, m\}$

Une chaîne de dépendance est une suite de tâches $X_{i_1,j_1}, \ldots, X_{i_k,j_k}$ telle que pour tout $2 \leq l \leq k$, $\mathcal{D}(i_l, j_l) = (i_{l-1}, j_{l-1})$.

Question 7 *Quelle est la longueur de la plus longue chaîne de dépendance dans les instances suivantes :* ***a)*** $\mathcal{O}(0, 8, 8)$ ***b)*** $\mathcal{O}(1, 1000, 50)$ ***c)*** $\mathcal{O}(2, 1000, 100)$ ***d)*** $\mathcal{O}(3, 1000, 200)$.

On souhaite maintenant déterminer le temps de terminaison minimal des tâches. On propose de modéliser ce problème en utilisant les contraintes de différences. Pour une instance $\mathcal{O}(t, m, n)$, on propose de définir un système de contraintes noté $\mathcal{S}(\mathcal{O}(t, m, n))$ dont les valuations correspondent aux exécutions de $\mathcal{O}(t, m, n)$ où toutes les tâches terminent avant la date T où T est une inconnue. Ce système est défini comme suit. On définit les inconnues $\nu(i, j)$ pour $(i, j) \in [1, m] \times [1, n]$ qui décrit la date d'exécution de la tâche $X_{i,j}$, et l'inconnue T.

$$\forall i \in [1, m], j \in [2, n], \nu(i, j) \geq \nu(i, j-1) + d_{i,j-1},$$
$$\forall i \in [1, m-1], j \in [1, n], \nu(i, j) \geq \nu(\mathcal{D}(i, j)) + d_{\mathcal{D}(i,j)},$$
$$\forall i \in [1, m], T \geq \nu(i, n) + d_{i,n}$$

Ainsi, les deux premières lignes définissent une exécution, et la troisième ligne contraint T à être supérieur à la date de terminaison de la dernière tâche de toutes les machines. On n'oubliera pas que toutes les dates d'exécution doivent être positives.

Question à développer pendant l'oral 5 *Étant donnée une instance du problème, on voudrait déterminer l'entier naturel minimal $T_{\min}$ tel qu'il existe une exécution qui termine avant la date $T_{\min}$. Décrire un algorithme pour calculer $T_{\min}$ et donner sa complexité. Combien d'appels votre algorithme fait-il à l'algorithme de Floyd-Warshall ? Pouvez-vous définir un algorithme qui appelle l'algorithme de Floyd-Warshall une seule fois ?*

Question 8 *Déterminer l'entier naturel minimal $T_{\min}$ tel qu'il existe une exécution qui termine avant la date $T_{\min}$, pour les instances suivantes :*
a) $\mathcal{O}(0, 5, 5)$ ***b)*** $\mathcal{O}(1, 10, 10)$ ***c)*** $\mathcal{O}(2, 15, 15)$ ***d)*** $\mathcal{O}(3, 30, 30)$.

Question à développer pendant l'oral 6 *Proposer un algorithme de complexité $O(mn)$ pour la question précédente.*

Question 9 *Déterminer l'entier minimal $T'_{\min}$ tel qu'il existe une exécution dans laquelle toutes les tâches de la machine $\lfloor m/2 \rfloor$ (c-à-d $X_{\lfloor m/2 \rfloor,1}, X_{\lfloor m/2 \rfloor,2}, \ldots, X_{\lfloor m/2 \rfloor,n}$) terminent avant la date $T'_{\min}$:*
a) $\mathcal{O}(0, 300, 300)$ ***b)*** $\mathcal{O}(1, 400, 400)$ ***c)*** $\mathcal{O}(2, 500, 500)$.

2 Planification temporisée

On s'intéresse maintenant à un problème de planification temporisée défini comme suit. On dispose d'un ensemble d'actions $X_1, \ldots, X_n$ qu'on peut effectuer. On souhaite choisir un sous-ensemble $S \subseteq \{X_1, \ldots, X_n\}$, appelé un plan, avec la contrainte $X_1, X_n \in S$. On dit que X_1 est l'action initiale, et X_n est l'action finale.

2.1 Contraintes de dates

On suppose que chaque action X_i est donnée avec une date d_i d'exécution imposée.
On voudrait trouver un plan S qui consiste à exécuter un sous-ensemble d'actions dans l'ordre croissant de leurs indices. On cherche ainsi une sous-suite croissante, c'est-à-dire $d_{\alpha_1}, \ldots, d_{\alpha_m}$ avec $1 = \alpha_1 < \alpha_2 < \ldots < \alpha_m = n$ et telle que $d_{\alpha_1} \leq d_{\alpha_2} \ldots \leq d_{\alpha_m}$. Notre but est de trouver la taille de la plus longue sous-suite croissante. On notera qu'une telle suite n'existe pas si $d_1 > d_n$.

On définit une instance $\mathcal{T}(t, n)$ du problème, sur n actions par le choix des dates, comme suit :

$$\forall i \in [1, n], d_i = u_{t,i}$$

Pour $1 \leq i \leq n$, on définit Q_i comme la longueur de la plus longue sous-suite de $d_1, d_2, \ldots, d_i$ qui se termine en d_i.

Question à développer pendant l'oral 7 *Proposer une définition de Q_i en fonction de $Q_1, \ldots, Q_{i-1}$.*

On définit l'ordre suivant entre deux suites $a = a_1 \ldots a_k$ et $b = b_1 \ldots b_k$ de même longueur comme suit. On dit que $a \prec b$ s'il existe $l \in [1, k]$ tel que $a_l < b_l$ et $\forall i \in [l+1, k], a_i = b_i$.

Pour tout n, soit $t_n \geq 1$ minimal pour lequel $\mathcal{T}(t_n, n)$ admet une sous-suite croissante. Parmi les sous-suites croissantes de longueur maximale, on voudrait calculer celle dont les indices définissent une suite maximale pour l'ordre $\prec$. Soit $h_n = \sum_{i=1}^{m} d_{\alpha(i)} \bmod 101$ où $d_{\alpha_1}, \ldots, d_{\alpha_m}$ est la plus longue sous-suite croissante de $\mathcal{T}(t_n, n)$ telle que $\alpha(1), \ldots, \alpha(m)$ est maximale pour l'ordre $\prec$.

Question 10 *Calculer h_n pour les instances suivantes.* ***a)*** $n = 100$ ***b)*** $n = 200$ ***c)*** $n = 400$.

2.2 Contraintes logiques

On considère maintenant un cas où il n'y a pas de date d_i associée aux actions mais les plans S considérés sont soumis à des contraintes logiques définies comme suit. On se donne un ensemble $\mathcal{C} \subseteq \{X_1, \ldots, X_n\}$ d'actions contraintes et associe, à chaque action $X_i \in \mathcal{C}$ une paire $(X_{\alpha(i)}, \neg X_{\beta(i)})$ avec $\alpha(i), \beta(i) \in [1, n]$.
On dit qu'un plan S est consistant si

$$\begin{array}{l} X_1 \in S, \\ X_n \in S, \\ \forall X_i \in \mathcal{C}, X_i \in S \Rightarrow X_{\alpha(i)} \in S \wedge X_{\beta(i)} \notin S. \end{array}$$

On définit les instances $\mathcal{P}(t, n)$ avec n actions $X_1, \ldots, X_n$. Pour tout $i \in [1, n]$, on a $X_i \in \mathcal{C}$ si et seulement si $u_{t,i} \equiv 0 \bmod 3$. De plus, pour tout $X_i \in \mathcal{C}$, $\alpha(i) = 1 + (u_{t,i,1} \bmod n)$ et $\beta(i) = 1 + (u_{t,i,2} \bmod n)$.

On dit qu'un plan S est un plan en bloc s'il s'écrit

$$S = \{X_1\} \cup \{X_i, X_{i+1}, \ldots, X_j\} \cup \{X_n\}$$

avec $(i, j) \in [2, n-1]^2$. On note que $\{X_1, X_n\}$ est un plan en bloc selon cette définition.

Question 11 *Pour chaque t_0 et n donnés, trouver le plus petit $t \in [0, 9]$ pour lequel $\mathcal{P}(t_0 + t, n)$ admet au moins un plan consistant en bloc, et pour ce t, écrire la paire $(t, |S|)$ où S est un plan en bloc consistant de taille maximale. Écrire $\emptyset$ s'il n'y a pas de tel plan consistant.*
a) $t_0 = 0, n = 100$. ***b)*** $t_0 = 10$, $n = 200$ ***c)*** $t_0 = 20$, $n = 400$.

Question à développer pendant l'oral 8 *Quelle est la complexité de votre algorithme ?*

On s'intéresse maintenant aux solutions générales.
On associe à chaque plan S le mot $w(S)$ sur l'alphabet $\Sigma = \{X_1, \ldots, X_n\}$, obtenu en concaténant les éléments S dans l'ordre croissant de leur indice. Par exemple, pour $S = \{X_1, X_3, X_4\}$, on a $w(S) = X_1X_3X_4$. On ordonne l'alphabet Σ selon les indices, c'est-à-dire $X_1 \prec X_2 \prec \ldots \prec X_n$.
On dit qu'un plan consistant S est minimal si le mot $w(S)$ est minimal pour l'ordre lexicographique sur Σ^* parmi tous les mots $w(S')$ pour lesquels S' est un plan consistant. Autrement dit, $w(S) = \min\{w(S') \mid S' \text{ plan consistant}\}$.

Question 12 *Pour t_0 et n donnés, calculer le plus petit $t \in [0, 9]$ pour lequel $\mathcal{S}(t_0 + t, n)$ admet un plan consistant. Si un tel t existe, calculer ce plan minimal consistant, et écrire le plus petit indice $i \in [1, n]$ qui n'y apparaît pas (et ∞ si tous les indices y apparaissent). Écrire $\emptyset$ s'il n'y a pas de tel plan. On n'oubliera pas que X_1 et X_n doivent faire partie d'un plan consistant.*
a) $t_0 = 0, n = 20$ ***b)*** $t_0 = 10, n = 100$ ***c)*** $t_0 = 20, n = 1000$.

Question à développer pendant l'oral 9 *Quelle est la complexité de votre algorithme ?*

2.3 Contraintes logiques et temporisées

On s'intéresse maintenant aux dates d'exécutions des actions choisies dans notre plan. On considère une valuation $v : [1, n] \to \mathbb{N}$ qui détermine la date d'exécution de chaque action. Par exemple, si $S = \{X_1, X_2\}$, alors X_1 est exécutée à $t = v(1)$, X_2 à $t = v(2)$. On se donne les contraintes supplémentaires suivantes pour le choix de ces dates dans un plan S :

$$\forall i, j \in S, v(i) - v(j) \leq \mathcal{E}(i, j), \tag{4}$$

où $\mathcal{E}(i, j) \in \mathbb{Z} \cup \{\infty\}$.
On dit qu'un plan S est réalisable s'il est consistant, et s'il existe une valuation v qui satisfait (4).
On construit les instances $\mathcal{P}'(n, t)$ en ajoutant à $\mathcal{P}(n, t)$ les contraintes

$$\mathcal{E}(i, j) = \begin{cases} 0 & \text{si } i = j, \\ u_{t,i,j} - \lfloor f/2 \rfloor & \text{sinon.} \end{cases}$$

Un plan réalisable est minimal s'il est minimal parmi les plans consistants et réalisables.

Question 13 *Pour t_0 et n donnés, calculer le plus petit $t \in [0, 9]$ pour lequel $\mathcal{S}(t_0 + t, n)$ admet un plan consistant et réalisable. Si un tel t existe, calculer ce plan minimal consistant réalisable, et écrire le plus petit indice $i \in [1, n]$ tel que X_i n'y apparaît pas (et ∞ si tous les indices y apparaissent). Écrire $\emptyset$ s'il n'y a pas de tel plan.*
a) $t_0 = 0, n = 20$ ***b)*** $t_0 = 10, n = 50$ ***c)*** $t_0 = 20, n = 250$.

Fiche réponse type: Contraintes et Planification

$\widetilde{u_0}$: 1

Question 1

a) 263

b) 205

c) 13

d) 2757

Question 2

a) $(1, 2, 30)$

b) $(1, 2, 86)$

c) $(5, 9, 149)$

Question 3

a) 1

b) 25

c) 1

Question 4

a) vrai

b) vrai

c) faux

Question 5

a) $(0, 96)$

b) $(0, 49)$

c) $(1, 15)$

Question 6

a) $(2, 3499)$

b) $(2, 7000)$

c) $(0, 60996)$

d) $(0, 36679)$

Question 7

a) 5

b) 20

c) 21

d) 20

Question 8

a) 18386

b) 65341

c) 129893

d) 429609

Question 9

a) 30211546

b) 57682686

c) 83705841

Question 10

a) 35

b) 88

c) 21

Question 11

a) $(0, 26)$

b) $(0, 8)$

c) $(1, 9)$

Question 12

a) $(0, 6)$

b) $(0, 13)$

c) $(0, 46)$

Question 13

a) $(3, 2)$

b) $(4, 2)$

c) $(2, 2)$

Fiche réponse: Contraintes et Planification

Nom, prénom, u_0: .....

Question 1

a)

b)

c)

d)

Question 2

a)

b)

c)

Question 3

a)

b)

c)

Question 4

a)

b)

c)

Question 5

a)

b)

c)

Question 6

a)

b)

c)

d)

Question 7

a)

b)

c)

d)

Question 8

a)

b)

c)

d)

Question 9

a)

b)

c)

Question 10

a)

b)

c)

Question 11

a)

b)

c)

Question 12

a)

b)

c)

Question 13

a)

b)

c)