Ensemble dominant dans les graphes.

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

Durée de l'épreuve : 3 heures 30 minutes

Juin/Juillet 2022

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.

1 Préliminaires

1.1 Rappel sur la récursion

La plupart des langages ont recours à une “pile d'appel” pour gérer la récursion. Cette pile d'appel a généralement une limite : une limite fixe de 1 000 appels récursifs en Python et une limite en taille de quelques méga-octets pour les langages comme OCaml ou C/C++ (ce qui généralement autorise plus de 100 000 appels récursifs). Attention, certaines questions de ce sujet manipulent des données suffisamment larges pour dépasser ces limites, surtout en Python !

1.2 Notations

On rappelle que pour deux entiers naturels a et b, $a \mod b$ désigne le reste de la division entière de a par b, c'est à dire l'unique entier r avec $0 \leq r < b$ tel que $a = k \times b + r$ pour $k \in \mathbb{N}$.
Ce sujet portant sur des graphes non orientés, chaque utilisation du mot graphe doit être entendue comme graphe non orienté sauf mention explicite du contraire. Un graphe est donc, pour ce sujet, un couple $G = \langle V, E \rangle$ où V est l'ensemble fini des sommets ou nœuds du graphe et E est l'ensemble des arêtes, chaque arête étant un ensemble de sommets de cardinal 2. Notez que cette définition interdit les boucles (un sommet n'est jamais relié à lui-même par une arête) ainsi que les arêtes multiples (deux nœuds ne peuvent être reliés que zéro ou une fois).
Par souci de simplicité, dans ce sujet, V est toujours constitué des nombres de 1 à n avec n le nombre de nœuds tandis que E est vu comme une liste de paires, chaque paire (x, y) de la liste aura $x < y$ et correspondra à l'arête $\{x, y\}$. Ainsi, un graphe est entièrement caractérisé par un nombre de nœuds et une liste d'arêtes.

1.3 Ensemble dominant d'un graphe

Étant donné un graphe $G = \langle V, E \rangle$, un sous-ensemble D des sommets ($D \subseteq V$) est dit dominant quand chaque sommet s de V est dominé, c'est à dire que s est dans D ou que s a un voisin dans D (un sommet dominé peut avoir plusieurs voisins dans D ou être dans D et avoir des voisins dans D). Dans ce sujet nous nous intéresserons au nombre dominant, c'est à dire à la taille minimale d'un ensemble dominant. L'objectif du sujet c'est de calculer ou d'approcher ce nombre dominant par des majorations.
Par exemple, dans le graphe de la figure 1, on a un nombre dominant de 5 car l'ensemble $\{1, 2, 4, 7, 10\}$ est dominant et aucun ensemble de taille 4 ne domine ce graphe. Noter que l'ensemble dominant n'est généralement pas unique, par exemple, le graphe de la figure 1 est aussi dominé, notamment, par $\{1, 2, 5, 8, 9\}$ mais le nombre dominant est unique car on s'intéresse au cardinal minimal d'un ensemble dominant.

1.4 Génération de nombres pseudo-aléatoires

Étant donné u_0, on définit la récurrence :

$$\forall t \in \mathbb{N},\ u_{t+1} = (900\,007 \times u_t) \mod 1\,000\,000\,007$$

L'entier u_0 vous est donné, et doit être recopié sur votre fiche réponse avec vos résultats. Une fiche réponse type vous est donnée en exemple, et contient tous les résultats attendus pour une valeur de u_0 différente de la vôtre (notée $\widetilde{u_0}$). Il vous est conseillé de tester vos algorithmes avec cet $\widetilde{u_0}$ et de comparer avec la fiche de résultat fournie. Pour chaque calcul demandé, avec le bon

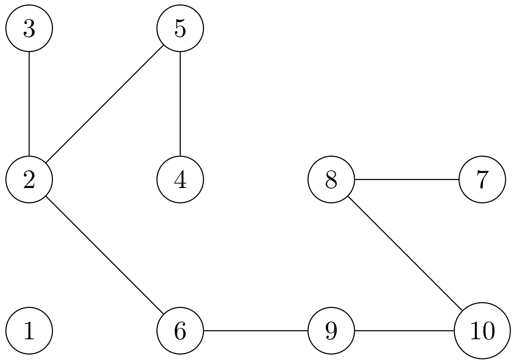


FIGURE 1 – Graphe correspondant à $G(10,4,4)$ pour $\widetilde{u_0}$

choix d'algorithme le calcul ne devrait demander qu'au plus de l'ordre de la seconde, jamais plus d'une minute.

Question 1 *Calculer les valeurs suivantes :*

a) $u_1 \ \ mod \ 1\,000$ ***b)*** $u_{12} \ \ mod \ 1\,000$ ***c)*** $u_{1234} \ mod \ 1\,000$ ***d)*** $u_{2345678} \ mod \ 1\,000$

2 Génération pseudo-aléatoire de graphes

2.1 Listes $P(n,m,k)$ et graphes $G(n,m,k)$

Étant une liste l de paires d'entiers ($l = (x_1,y_1),\ldots,(x_n,y_n)$) et un entier k, on définit la fonction filtre $f_k(l)$ qui renvoie la liste des paires (x,y) de l telles que $u_{17\times y+x} \mod k = 0$. Étant donnés n et m, on définit $L(n,m)$ comme étant la liste de paires (x,y) avec $1 \leq x < y \leq n$ et $y \leq x+m$. On définit alors la liste $P(n,m,k)$ comme étant la liste $f_k(L(n,m))$ et le graphe $G(n,m,k)$ comme étant le graphe à n nœuds dont la liste d'arêtes est $P(n,m,k)$.

Question 2 *Calculer le nombre d'arêtes des graphes suivants :*

a) $G(123,10,5)$ ***b)*** $G(1\,234,10,5)$ ***c)*** $G(12\,345,12,6)$ ***d)*** $G(54\,321,22,8)$

Question à développer pendant l'oral 1 *Décrire la structure de donnée que vous avez choisie pour représenter les graphes. En particulier, vous devez décrire la complexité des opérations suivantes :*

— *l'occupation mémoire (en ordre de grandeur pour n nœuds et e arêtes),*
— *la complexité de récupérer la liste des voisins d'un nœud, ordonnée par indices (si a et b sont des voisins alors a apparaît avant b dans la liste quand $a < b$).*

2.2 Somme de contrôle d'un graphe

Pour un graphe $G = \langle V, E\rangle$, on définit sa somme de contrôle $SC(G)$ à l'aide de la formule suivante :

$$SC(G) = \left(\sum_{x \in V} \sum_{y | \{x,y\} \in E} (x+y)^2 + y \right) \mod 1\,000\,000$$

Question 3 *Calculer la somme de contrôle des graphes suivants :*

a) $G(123, 10, 5)$ ***b)*** $G(1\,234, 10, 5)$ ***c)*** $G(12\,345, 12, 6)$ ***d)*** $G(54\,321, 22, 8)$

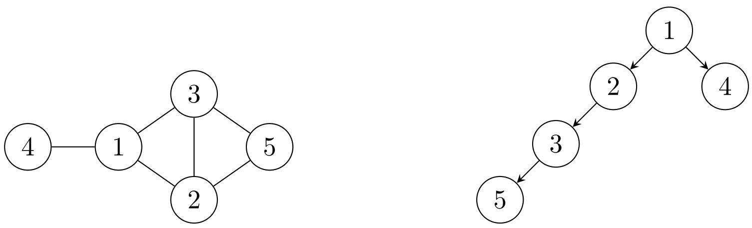
2.3 Génération d'arbre à partir de graphes

Attention, cette section 2.3 concerne la génération d'arbre et n'est utile que pour les questions 4 et 8. Si vous bloquez, n'hésitez pas à sauter cette section et à vous concentrer d'abord sur la question 5.

Étant donné un graphe G, l'ensemble $A(u, S)$ est défini comme l'ensemble des nœuds v tels que v est accessible depuis le nœud u en évitant un ensemble de nœuds S, c'est à dire qu'il y a un chemin de u à v où aucun des nœuds du chemin n'est dans S. Formellement, si $N(u)$ renvoie les voisins de u, on a :

$$A(u,S) = \begin{cases} \emptyset & \text{quand } u \in S \\ \{u\} \displaystyle\bigcup_{v \in N(u)} A(v, S \cup \{u\}) & \text{sinon} \end{cases}$$

Soit u un nœud de G et S un ensemble de nœuds nous allons définir l'arbre $T(u, S)$ enraciné en u. Pour cela, on note $v_1, \ldots, v_k$ les voisins de u dans G avec $v_1 < \cdots < v_k$ et on pose $S_0 = S \cup \{u\}$ puis $S_{j+1} = S_j \cup A(v_{j+1}, S_j)$. Pour $u \in S$ alors $T(u, S)$ est l'arbre vide. Pour $u \notin S$, l'arbre $T(u, S)$ est l'arbre enraciné en u dont les sous-arbres sont les $T(v_i, S_{i-1})$.
Pour finir, étant donné un graphe G, on définit $\text{Arbre}(G) = T(1, \{\})$. Dans la figure suivante on voit à gauche un graphe G et à droite l'arbre $\text{Arbre}(G)$.



Notez que les arbres enracinés sont généralement vus comme des graphes orientés mais nous n'utiliserons l'orientation que pour présenter la construction des arbres et pour le calcul de la somme de contrôle. Dès qu'il s'agira de trouver un ensemble dominant pour un arbre, celui-ci sera vu comme un graphe non orienté, c'est à dire qu'un sommet sera dominé par D s'il est dans D ou si son parent est dans D ou si l'un de ses enfants est dans D.
Pour un arbre $\mathcal{T}$, la somme $SC(\mathcal{T})$ de contrôle de $\mathcal{T}$ est définie comme $\displaystyle\sum_{x \text{ parent de } y} (x+y)^2 + y \mod 1\,000\,000$. Notez que cette formule ressemble à celle des graphes mais elle est "orientée" : dans la version graphe, une arête $\{x, y\}$ est comptée deux fois tandis qu'ici elle n'est comptée qu'une seule fois.

Question 4 *Calculer la somme de contrôle pour les arbres suivants :*

a) $\mathit{Arbre}(G(123, 10, 5))$ **b)** $\mathit{Arbre}(G(1\,234, 10, 5))$

c) $\mathit{Arbre}(G(12\,345, 12, 6))$ **d)** $\mathit{Arbre}(G(54\,321, 22, 8))$

3 Majoration du nombre dominant par une heuristique

Nous étudions maintenant un algorithme pour calculer un ensemble dominant. Cet algorithme, que l'on nommera $\mathcal{A}_1$, fonctionne de la façon suivante : l'algorithme définit $D_0 = \emptyset$ et itère ensuite sur i de 1 à n. Pour chaque i, si le nœud i est dominé par D_{i-1} alors $D_i = D_{i-1}$ et sinon $D_i = \{i\} \cup D_{i-1}$. Le résultat de l'algorithme est D_n.

Question à développer pendant l'oral 2 *Justifier que cet algorithme produit un ensemble dominant mais qu'il n'est pas toujours de taille minimale. Est-ce que la taille de l'ensemble dominant produit par l'algorithme reste proche du nombre dominant pour tous les graphes ? Justifier.*

Question 5 *Calculer la taille de l'ensemble renvoyé par $\mathcal{A}_1$ sur les graphes suivants :*

a) $G(123, 10, 5)$ **b)** $G(1\,234, 10, 5)$ **c)** $G(12\,345, 12, 6)$ **d)** $G(54\,321, 22, 8)$

Question à développer pendant l'oral 3 *Décrire la complexité de votre algorithme en fonction du nombre de nœuds et du nombre d'arêtes des graphes manipulés. Pour le calcul de complexité, le temps de création du graphe ne sera pas pris en compte.*

4 Majoration du nombre dominant par la décomposition en composantes connexes

La décomposition d'un graphe en composantes connexes est la partition des sommets d'un graphe en des ensembles $V_1, \ldots, V_k$ tels que deux sommets u et v sont dans le même ensemble V_i si et seulement si, il y a un chemin entre u et v. Les V_i sont les composantes connexes du graphe et chaque V_i forme un sous graphe que l'on peut étudier individuellement. L'entier k est appelé le nombre de composantes connexes (on suppose qu'aucun des V_i n'est vide). Enfin on dit qu'un sommet est isolé s'il est seul dans sa composante connexe.

Question à développer pendant l'oral 4 *Montrer que tout graphe à n nœuds sans sommet isolé admet un ensemble dominant de taille au plus $n/2$. Les propositions prouvant seulement la borne de $(n+1)/2$ obtiendront une partie des points.*

Nous étudions alors un algorithme, que l'on nommera $\mathcal{A}_2$ pour borner le nombre dominant. L'algorithme $\mathcal{A}_2$ fonctionne de la façon suivante : il calcule d'abord les tailles $T_1 \ldots T_k$ des diverses composantes connexes puis renvoie $\sum_i t(T_i)$ où $t(1) = 1$ et $t(\ell) = \lfloor \ell/2 \rfloor$ pour $\ell > 1$.

Question 6 *Calculer la valeur renvoyée par $\mathcal{A}_2$ sur les graphes suivants :*

a) $G(123, 10, 5)$ **b)** $G(1\,234, 10, 5)$ **c)** $G(12\,345, 12, 6)$ **d)** $G(54\,321, 22, 8)$

Question à développer pendant l'oral 5 *Décrire la complexité de votre algorithme en fonction du nombre de nœuds et du nombre d'arêtes des graphes manipulés. Pour le calcul de complexité, le temps de création du graphe ne sera pas pris en compte.*

5 Majoration du nombre dominant par une autre heuristique

L'algorithme $\mathcal{A}_1$ que nous avons vu plus haut traite les sommets dans l'ordre de leurs indices (d'abord le sommet 1, puis le 2, etc.). L'algorithme $\mathcal{A}_3$ que nous introduisons maintenant, fonctionne avec un score de dominance. Pour un ensemble D de sommets, le score $s(D)$ est le nombre de sommets dominés par D. Au lancement de l'algorithme, on a $D_0 = \emptyset$ et ensuite, tant que $s(D_i) \neq n$ l'algorithme choisit le sommet u tel que $s(D_i \cup \{u\})$ est maximal et définit $D_{i+1} = D_i \cup \{u\}$. En cas d'égalité entre plusieurs sommets qui donnent le meilleur score, l'algorithme choisit celui qui a le plus petit indice.

Question 7 *Calculer la taille de l'ensemble renvoyé par $\mathcal{A}_3$ sur les graphes suivants :*

a) $G(123, 10, 5)$ ***b)*** $G(1\,234, 10, 5)$ ***c)*** $G(9876, 1234, 6)$ ***d)*** $G(54\,321, 22, 8)$

Attention, le graphe **c)** diffère des questions précédentes !

Question à développer pendant l'oral 6 *Décrire la complexité de votre algorithme en fonction du nombre de nœuds et du nombre d'arêtes des graphes manipulés. Pour le calcul de complexité, le temps de création du graphe ne sera pas pris en compte.*

6 Calcul exact du nombre dominant

Dans cette partie, nous allons trouver le véritable nombre dominant. On note $nd(G)$ le nombre dominant du graphe G.

6.1 Cas d'un arbre

Attention, on rappelle que les arbres manipulés sont des graphes non orientés.
Pour $\mathcal{T}$ un arbre, on définit : nombreDominantAvecRacinePrise($\mathcal{T}$) la taille minimale d'un ensemble dominant pour $\mathcal{T}$ qui contient la racine et nombreDominantSaufRacine($\mathcal{T}$), la taille minimale d'un ensemble qui domine tous les nœuds de $\mathcal{T}$ sauf éventuellement la racine.

Question à développer pendant l'oral 7 *Étudier la relation entre les valeurs de $nd(\mathcal{T})$, nombreDominantAvecRacinePrise($\mathcal{T}$), nombreDominantSaufRacine($\mathcal{T}$) et les valeurs de ces fonctions sur les sous-arbres de $\mathcal{T}$. En déduire un algorithme pour calculer $nd(\mathcal{T})$, quelle est sa complexité ?*

Question 8 *Calculer le nombre dominant pour les arbres suivants :*

a) *Arbre*$(G(123, 10, 5))$ ***b)*** *Arbre*$(G(1\,234, 10, 5))$
c) *Arbre*$(G(12\,345, 12, 6))$ ***d)*** *Arbre*$(G(54\,321, 22, 8))$

6.2 Cas des petits graphes

Dans cette sous-partie, il vous est demandé d'écrire un algorithme qui calcule le nombre dominant pour un graphe quelconque. Pour cela, il faut générer tous les sous-ensembles de sommets et tester si ceux-ci dominent ou non le graphe. Attention les cas **e)** et **f)** requièrent une exploration optimisée de l'ensemble des sous-ensembles de sommets.

Question 9 *Calculer le nombre dominant pour les graphes suivants :*

a) $G(10, 5, 2)$ ***b)*** $G(12, 3, 2)$ ***c)*** $G(14, 4, 2)$

d) $G(18, 3, 2)$ ***e)*** $G(45, 20, 2)$ ***f)*** $G(60, 25, 2)$

6.3 Cas de nos graphes générés avec un petit paramètre m

Les graphes que l'on génère dans ce sujet ont une forme bien particulière quand le paramètre m est petit même si n est grand. De la même façon que dans un arbre un nœud ne peut être dominé que par son parent, par lui-même ou par l'un de ses descendants, dans nos graphes un nœud x ne peut être dominé que par un nœud y avec $x - m \leq y \leq x + m$.

Question 10 *Calculer le nombre dominant pour les graphes suivants :*

a) $G(2\,000, 8, 4)$ ***b)*** $G(3\,000, 7, 5)$

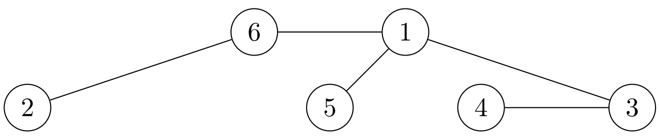
c) $G(4\,000, 6, 6)$ ***d)*** $G(5\,000, 6, 6)$

Question à développer pendant l'oral 8 *Détailler votre algorithme et donner sa complexité.*

6.4 Cas d'une classe spéciale de graphes

Pour cette section, nous allons générer un nouveau type de graphe. Le graphe $C(n)$ est défini comme le graphe à n nœuds dont les arêtes sont $\{i, 1 + (u_i \mod n)\}$ pour tout $1 \leq i \leq n$ avec $1 + (u_i \mod n) \neq i$.

Noter que $C(n)$ contient toujours n nœuds mais pas forcément n arêtes. Il est possible, par exemple, que $1 + (u_i \mod n) = i$ ou que l'on ait des paires i, j telles que $j = 1 + (u_i \mod n)$ et $i = 1 + (u_j \mod n)$. Pour $\widetilde{u_0}$, voici $C(6)$, qui a 5 arêtes :



Question 11 *Calculer les valeurs suivantes :*

a) $nd(C(10))$ ***b)*** $nd(C(100))$

c) $\displaystyle\sum_{5 \leq i \leq 1\,000} nd(C(i))$ ***d)*** $\displaystyle\sum_{100\,000 \leq i \leq 100\,100} nd(C(i))$

Question à développer pendant l'oral 9 *Détailler votre algorithme, donner sa complexité et justifier sa correction.*

Fiche réponse type : Ensemble dominant dans les graphes.

$\widetilde{u_0}$: 503

Question 1

a) 521

b) 165

c) 668

d) 896

Question 2

a) 215

b) 2398

c) 24476

d) 148185

Question 3

a) 348194

b) 362363

c) 997270

d) 48746

Question 4

a) 340200

b) 305976

c) 959830

d) 959765

Question 5

a) 57

b) 510

c) 5158

d) 19403

Question 6

a) 63

b) 626

c) 6246

d) 27242

Question 7

a) 36

b) 342

c) 83

d) 11800

Question 8

a) 40

b) 413

c) 4155

d) 18253

Question 9

a) 3

b) 5

c) 4

d) 6

e) 5

f) 5

Question 10

a) 473

b) 994

c) 1658

d) 2080

Question 11

a) 4

b) 35

c) 180601

d) 3638480

Fiche réponse : Ensemble dominant dans les graphes.

Nom, prénom, u_0 : ....

Question 1

a)

b)

c)

d)

Question 2

a)

b)

c)

d)

Question 3

a)

b)

c)

d)

Question 4

a)

b)

c)

d)

Question 5

a)

b)

c)

d)

Question 6

a)

b)

c)

d)

Question 7

a)

b)

c)

d)

Question 8

a)

b)

c)

d)

Question 9

a)

b)

c)

d)

e)

f)

Question 10

a)

b)

c)

d)

Question 11

a)

b)

c)

d)