

Toute réponse non justifiée n'apportera pas de point.

Vous pouvez utiliser le résultat d'une question même si elle n'a pas été traitée.

Le barème est indicatif.

Exercice 1 : NP-complet ou NL-complet

8 Points

Pour chacun des problèmes ci-dessous, **prouver** qu'il est soit **NP-complet**, soit **NL-complet**, soit dans **L**, en justifiant.
Tous les objets donnés en entrée sont finis.

1) COUVERTURE DE CYCLES

Entrée : Un graphe non dirigé G et un entier k .

Question : Existe-t-il m sommets v_1, \dots, v_m de G , avec
– $m \leq k$, et
– Tout cycle de G passe par au moins l'un des sommets v_i .

On rappelle qu'un *cycle* dans un graphe **dirigé** est un chemin $s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_{n-1} \rightarrow s_n$ dans lequel $s_0 = s_n$. Un *chemin* est *simple* lorsque tous les sommets par lesquels il passe sont distincts deux à deux, sauf éventuellement le sommet de départ et le sommet d'arrivée qui peuvent être égaux (le chemin est dans ce cas un cycle simple).

2) PLUS LONG CHEMIN

Entrée : Un graphe **dirigé** G , deux sommets s, t de G et un entier $k > 0$.

Question : Existe-t-il dans G un chemin simple de longueur **supérieure** à k , allant de s à t ?

3) PLUS COURT CHEMIN

Entrée : Un graphe **dirigé** G , deux sommets s, t de G et un entier $k > 0$.

Question : Existe-t-il dans G un chemin de longueur **inférieure** à k , allant de s à t ?

4) Un graphe non dirigé G est *connexe* si pour tous sommets s, t , il existe un chemin de s à t . Un *arbre* est un graphe non dirigé, connexe sans cycle. Enfin, une feuille d'un arbre est un sommet qui a un unique voisin.

ARBRE COUVRANT À PEU DE FEUILLES

Entrée : Un graphe non dirigé G et un entier $k > 0$.

Question : Existe-t-il un sous-ensemble E d'arêtes de G formant un arbre contenant tous les sommets de G , et ayant au maximum k feuilles ?

5) CYCLE SIMPLE DE POIDS 0

Entrée : Un graphe connexe **dirigé** G et une fonction qui à chaque arête de G associe un entier dans \mathbb{Z} , appelé *poids* de l'arête.

Question : Y a-t-il dans G un cycle simple dont la somme des poids des arêtes est 0 ?

6) Un **semigroupe** est un ensemble S muni d'une multiplication associative $(x, y) \mapsto xy$, c'est-à-dire que pour tous $x, y, z \in M$, on a $(xy)z = x(yz)$.

SEMIGROUPE COMMUTATIF

Entrée : Un semigroupe S donné par sa table de multiplication.

Question : Le semigroupe S est-il commutatif, c'est-à-dire, a-t-on $xy = yx$ pour tous éléments x, y de S ?

7) Si S est un semigroupe et X un sous-ensemble d'éléments de S , le *semigroupe engendré par X* est l'ensemble de tous les produits de la forme $x_1 x_2 \dots x_n$ avec $n > 0$ et où chaque x_i est un élément de X (un même élément peut apparaître plusieurs fois dans ce produit).

APPARTENANCE À UN SOUS-SEMIGROUPE

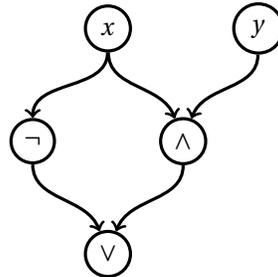
Entrée : Un semigroupe S donné par sa table de multiplication, un sous-ensemble X de S , et un élément x de S .

Question : L'élément x appartient-il au semigroupe engendré par X ?

Un **circuit Booléen** est un **graphe dirigé sans cycle** dont les sommets sont appelés des *portes*. Les portes sont étiquetées soit par des noms de variables, soit par des connecteurs logiques (\wedge pour **Et**, \vee pour **Ou** et \neg pour **Non**). Un circuit doit satisfaire les propriétés suivantes :

- Les portes qui n'ont pas d'arête entrante (appelées *portes d'entrée*) sont étiquetées par un nom de variable.
- Les autres portes sont étiquetées par des connecteurs logiques (\wedge , \vee ou \neg). Une porte étiquetée par \neg a une unique arête entrante, et une porte étiquetée par \wedge ou \vee a exactement 2 arêtes entrantes.
- Il y a une unique porte sans arête sortante, appelée la *porte résultat*. Les autres portes peuvent avoir un nombre arbitraire d'arêtes sortantes.

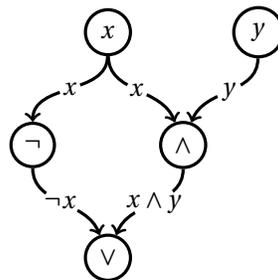
Un exemple de circuit Booléen est donné ci-dessous. La porte résultat est étiquetée \vee , et les portes d'entrée par les noms de variables x et y . La porte étiquetée par x a deux arêtes sortantes (l'une vers la porte \neg , l'autre vers la porte \wedge).



Chaque porte calcule une formule de logique propositionnelle. Sauf pour la porte résultat, cette formule est ensuite transmise le long de toutes ses arêtes qui sortent de la porte. Intuitivement, une porte effectue sur ses arêtes entrantes l'opération logique correspondant à son étiquette :

- Une porte d'entrée étiquetée par x calcule la formule " x ".
- Une porte étiquetée par \neg , et dont l'unique arête entrante transmet une formule φ , calcule $\neg\varphi$.
- Une porte étiquetée par \vee , et dont les deux arêtes entrantes transmettent des formules φ_1 et φ_2 , calcule $\varphi \vee \varphi_2$.
- Une porte étiquetée par \wedge , et dont les deux arêtes entrantes transmettent des formules φ_1 et φ_2 , calcule $\varphi \wedge \varphi_2$.

On dit que le circuit calcule la formule qui est calculée par sa porte résultat. La figure ci-dessous indique les formules transmises par les portes (sauf la porte résultat) de l'exemple précédent. Le circuit calcule donc la formule $\neg x \vee (x \wedge y)$.



1) On considère d'abord le problème SATISFIABILITÉ DE CIRCUIT suivant :

Entrée : Un circuit Booléen C .

Question : La formule calculée par le circuit C est-elle satisfiable?

Ce problème est-il **NP**-complet? Justifier.

2) Dans la suite, on représente les Booléens par des entiers (**Vrai** par 1 et **Faux** par 0). On considère un circuit Booléen dans lequel on donne des valeurs aux variables apparaissant dans les portes d'entrée. Par exemple, le circuit précédent évalué pour $x = 1$ et $y = 0$ calcule la valeur 0. Par contre, pour les autres valeurs du couple (x, y) , il calcule la valeur 1.

On considère donc le problème ÉVALUATION DE CIRCUIT suivant :

Entrée : Un circuit Booléen C et pour chaque variable présente dans C , une valeur 0 ou 1 pour cette variable.

Question : La formule calculée par le circuit C vaut-elle 1 lorsque chaque variable est remplacée par sa valeur?

Montrer que ÉVALUATION DE CIRCUIT est dans **P**.

3) Expliquer pourquoi ÉVALUATION DE CIRCUIT est **P**-complet.

Indication

Réduisez directement un problème arbitraire de la classe **P**, en partant d'une machine de Turing M déterministe qui décide ce problème en temps polynomial. Il s'agit donc, à partir d'une entrée x de M , de construire un circuit qui lit dans ses portes d'entrée les $|x|$ bits de x , et calcule 1 si et seulement si le mot x est accepté par M .
 Comme il s'agit d'une réduction générique, il est possible de s'inspirer de la preuve du théorème de Cook-Levin montrant que **SAT** est **NP**-complet : vous pouvez considérer le tableau espace-temps des configurations de M , chaque symbole codé en binaire, et analyser comment un bit à l'étape $i + 1$ est produit à partir de ceux de l'étape i .
 Enfin, n'oubliez pas de vérifier la complexité de votre réduction.

4) On considère maintenant la restriction suivante du problème ÉVALUATION DE CIRCUIT, appelée ÉVALUATION DE CIRCUIT MONOTONE :

Entrée : Un circuit Booléen C sans porte étiquetée \neg , et pour chaque variable présente dans C , une valeur 0 ou 1 pour cette variable.

Question : Le circuit C calcule-t-il la valeur 1?

Montrer que ÉVALUATION DE CIRCUIT MONOTONE est encore **P**-complet.

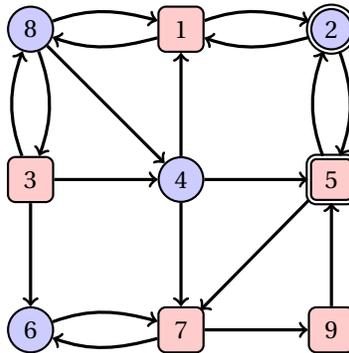
5) Déterminer la plus petite valeur de $k > 0$ telle que le problème ÉVALUATION DE CIRCUIT soit résoluble en temps $O(n^k)$.

6) Est-il possible qu'un problème ait une complexité linéaire et soit **P**-complet? Expliquer votre réponse.

On s'intéresse maintenant à la résolution algorithmique des jeux d'accessibilité, et à leur complexité. Un tel jeu se joue à deux joueurs, nommés J_0 et J_1 . Une partie se joue sur un graphe dirigé $G = (S, A)$ (S est l'ensemble des sommets, A celui des arêtes) ayant les propriétés suivantes :

- L'ensemble S est partitionné en deux sous-ensembles disjoints S_0 et S_1 , c'est-à-dire que $S = S_0 \cup S_1$, avec $S_0 \cap S_1 = \emptyset$.
- Tout sommet de G a au moins une arête sortante.

Un exemple de tel graphe avec les sous-ensembles S_0, S_1 est donné sur la figure suivante. Les sommets de $S_0 = \{2, 4, 6, 8\}$ sont dessinés dans des cercles en bleu, et ceux de $S_1 = \{1, 3, 5, 7, 9\}$ dans des carrés en rouge.



On se donne aussi un sous-ensemble de sommets $But \subseteq S$. Sur la figure, $But = \{2, 5\}$ (sommets doublement entourés).

Une partie se déroule de la façon suivante. Un jeton est initialement placé sur un des sommets de G . Le jeu se déroule en tours. À chaque tour, si le jeton est sur un sommet s de S_0 , alors J_0 joue, et sinon J_1 joue. Le joueur qui joue doit déplacer le jeton sur un sommet t tel que (s, t) est une arête. Dans l'exemple, si le jeton est sur le sommet 8, c'est J_0 qui joue (car $8 \in S_0$) et il doit déplacer le jeton sur 1, 3 ou 4 (car les arêtes partant de 8 sont $(8, 1)$, $(8, 3)$ et $(8, 4)$).

Le joueur J_0 gagne la partie si le jeton arrive, à n'importe quel moment, en un sommet appartenant à But . Si J_0 n'arrive pas à guider le jeton sur un sommet de But , alors J_1 gagne. Par exemple, à partir du sommet 1, J_0 a une stratégie gagnante : si J_1 déplace le jeton sur 2, J_0 gagne directement. Sinon, J_1 déplace le jeton sur 8, et J_0 le déplace ensuite sur 4, puis sur 5.

7) On considère les ensembles de sommets suivants (on rappelle que A est l'ensemble des arêtes du graphe) :

$$X_0 = But, \quad \text{et} \quad X_{n+1} = X_n \cup \left\{ s \in S_0 \mid \text{il existe } t \text{ tel que } (s, t) \in A \text{ et } t \in X_n \right\} \cup \left\{ s \in S_1 \mid \text{pour tout } t \text{ tel que } (s, t) \in A, \text{ on a } t \in X_n \right\}.$$

Calculer les ensembles X_n pour $n = 0, 1, 2, 3, 4, 5$ sur l'exemple précédent.

- 8) On note $X = \bigcup_{n \geq 0} X_n$. Montrer que X se calcule en temps polynomial (en fonction de la taille de G) et que J_0 a une stratégie pour gagner toute partie commençant sur un sommet de l'ensemble X .
- 9) Montrer que le joueur J_1 a une stratégie pour gagner toute partie commençant sur un sommet de l'ensemble $S \setminus X$.
- 10) On considère le problème JEU D'ACCESSIBILITÉ suivant :
- Entrée** : Un graphe G satisfaisant les conditions ci-dessus avec des sous-ensembles de sommets S_0, S_1 , But, et un sommet s de G .
- Question** : Le joueur J_0 a-t-il une stratégie gagnante si le jeu débute avec le jeton sur s ?
- Montrer que ce problème est dans **P**.
- 11) Montrer que le problème JEU D'ACCESSIBILITÉ est **P**-complet.

 **Indication**

Vous pouvez utiliser, même si vous ne l'avez pas faite, le résultat d'une question précédente. On a montré que le problème ÉVALUATION DE CIRCUIT MONOTONE est **P**-complet.

Exercice 3 : Somme d'entiers

4 Points

On considère le problème **Somme d'Entiers** suivant :

ENTRÉE : Des entiers positifs v_1, v_2, \dots, v_n et un entier S .

QUESTION : Existe-t-il une sous-suite $1 \leq i_1 < i_2 < \dots < i_p \leq n$ telle que $v_{i_1} + v_{i_2} + \dots + v_{i_p} = S$?

- 1) On suppose que la séquence v_1, \dots, v_n est déjà triée $v_1 \leq v_2 \leq \dots \leq v_n$. Pour le problème **Somme d'Entiers**, proposer un algorithme qui utilise une mémoire de taille $O(S)$ et dont le temps de calcul est $O(n \cdot S)$.
- 2) Il est connu que le problème **Somme d'Entiers** est **NP**-complet. Pourquoi l'existence de l'algorithme de la question 1 ne contredit-elle pas la **NP**-complétude du problème ?

On veut maintenant montrer que le problème **Somme d'Entiers** est en effet **NP**-difficile. On va réduire 3-SAT à **Somme d'Entiers**. Il faut donc décrire un algorithme polynomial qui, étant donné une formule 3-SAT,

$$\varphi = (\ell_{1,1} \vee \ell_{1,2} \vee \ell_{1,3}) \wedge \dots \wedge (\ell_{m,1} \vee \ell_{m,2} \vee \ell_{m,3})$$

produit une suite d'entiers v_1, \dots, v_n et un entier S tels que S est la somme d'un sous-ensemble d'entiers parmi v_1, \dots, v_n si et seulement si φ est satisfiable. Nous présentons une idée de construction sur un exemple. On considère la formule :

$$\varphi = \underbrace{(x_1 \vee x_2 \vee \neg x_3)}_{c_1} \wedge \underbrace{(\neg x_1 \vee x_2 \vee \neg x_3)}_{c_2}.$$

On introduit d'abord des entiers $v_1, \bar{v}_1, v_2, \bar{v}_2, v_3, \bar{v}_3$ qui représentent les occurrences de chaque littéral (positif ou négatif) dans les clauses c_1 et c_2 de φ . On donne les valeurs de $v_1, \bar{v}_1, v_2, \bar{v}_2, v_3, \bar{v}_3$ en utilisant leur représentation décimale, sur 5 chiffres, comme indiqué dans le tableau suivant (ainsi par exemple, v_2 vaut l'entier "mille onze") :

	x_1	x_2	x_3	c_1	c_2
v_1	1	0	0	1	0
\bar{v}_1	1	0	0	0	1
v_2	0	1	0	1	1
\bar{v}_2	0	1	0	0	0
v_3	0	0	1	0	0
\bar{v}_3	0	0	1	1	1

- 3) Que peut-on dire d'un sous-ensemble de $\{v_1, \bar{v}_1, v_2, \bar{v}_2, v_3, \bar{v}_3\}$ dont la somme est de la forme $S = 111--$, où $--$ sont deux chiffres différents de 0 (c'est-à-dire ≥ 1 , on est en base 10) ?

Pour terminer l'exemple, on ajoute deux autres entiers pour chaque clause :

	x_1	x_2	x_3	c_1	c_2
w_1	0	0	0	1	0
w_2	0	0	0	1	0
w_3	0	0	0	0	1
w_4	0	0	0	0	1

- 4) Montrez qu'il existe un sous-ensemble de $\{v_1, \bar{v}_1, v_2, \bar{v}_2, v_3, \bar{v}_3, w_1, w_2, w_3, w_4\}$ dont la somme est $S = 11133$ si et seulement si φ est satisfiable.
- 5) Généraliser la construction précédente en formalisant la réduction.