UE COMPLEX.
M1 Informatique.

Examen du 10 décembre 2020.

Seule une feuille A4 portant sur les cours et les TD est autorisée, tout autre document est interdit. Téléphones portables éteints et rangés dans vos sacs. Le barème est indicatif et est susceptible d'être modifié. Toutes les réponses doivent être justifiées.

Exercice 1 (2 points)

Répondre aux questions à choix multiples suivantes (pour chaque question, une ou plusieurs réponses peuvent être correctes). Chaque mauvaise réponse retranchera 0.25 point au total de l'exercice (qui restera tout de même au minimum 0). Une absence de réponse ne retranchera pas de point au total. Pour chacune des questions de cet exercice, <u>on supposera que $P \neq NP$.</u>

Questions	**Réponses**
1. Soit A un problème de décision. Si le problème SAT se réduit en temps polynomial au problème A, alors il n'est pas possible d'obtenir un algorithme polynomial résolvant le problème A.	■ Vrai.
	☐ Faux.
2. S'il existe un schéma d'approximation polynomial pour un problème de minimisation NP-difficile, alors il existe un algorithme polynomial 3-approché pour ce même problème.	■ Vrai.
	☐ Faux.
3. Cocher la ou les bonne(s) réponse(s) parmi les suivantes. Un algorithme de branch and bound :	☐ Permet de résoudre en temps polynomial un problème NP-difficile.
	■ Permet en général de résoudre un problème d'optimisation combinatoire plus rapidement qu'une énumération exhaustive des solutions réalisables de ce problème.
	■ Divise un problème en au moins deux nouveaux sous-problèmes.
4. Quel type de données abstrait est utilisé pour implémenter un algorithme de branch and bound avec parcours en profondeur de l'arbre de recherche ?	☐ Une file.
	■ Une pile.
	☐ Une liste chaînée.
	☐ Un tableau.

Exercice 2 (4 points)

Étant donné un mot commençant par un symbole # et constitué ensuite de A et de B, on souhaite savoir si ce mot contient plus de A que de B. Décrire en quelques phrases puis formellement (en faisant un diagramme d'états notamment) une machine de Turing résolvant de problème. Celle-ci devra terminer dans l'état d'acceptation si et seulement si le mot commence par un # et est ensuite suivi de A et de B, avec un nombre de A strictement supérieur au nombre de B. Ainsi le mot $\#ABBAA$ sera accepté tandis que les mots $ABAA$ et $\#ABAB$ seront rejetés.

Exercice 3 (14 points)

Etant donné un graphe orienté $R = (V, A)$, un ensemble d'arcs $A' \subseteq A$ est dit *sans circuit* si le graphe R restreint aux arcs de A' (le graphe (V, A')) ne contient pas de circuit. Considérons par exemple le graphe R à gauche de la figure 1. Il contient des circuits (par exemple (v_1, v_2, v_4, v_1)). L'ensemble $A' = \{((v_1, v_3), (v_2, v_3), (v_2, v_4), (v_4, v_1))\}$ est sans circuit (voir la partie droite de la figure).

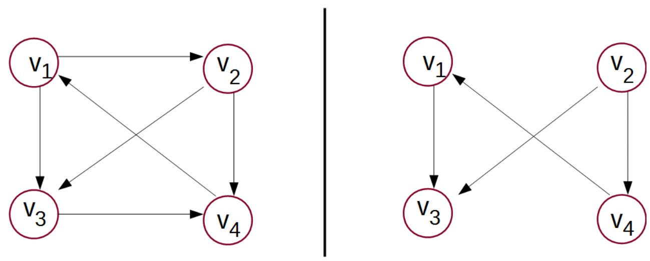

FIGURE 1 – Le graphe R (à gauche), et l'ensemble d'arcs sélectionnés (à droite)

On considère dans cet exercice le problème SGSS (pour sous-graphe sans circuit) où, étant donné un graphe orienté R, on cherche un ensemble d'arcs sans circuit de taille maximale. On considère aussi la version décision SGSS-D de ce problème définie de la manière suivante.

— Entrée : un graphe orienté R et un entier k.
— Question : existe-t-il dans R un ensemble d'arcs sans circuit de taille au moins k ?

Question 1 (1.5/14) — On considère le graphe R de la figure 1 (partie gauche), où l'on a donné une solution de taille 4. Donner une solution optimale du problème SGSS (on ne demande pas de justification).

On peut supprimer uniquement l'arc (v_4, v_1), et obtenir un ensemble sans circuit de 5 arcs.

Question 2 (4.5/14) — Soit $R = (V, A)$ un graphe orienté, avec $V = \{v_1, v_2, \ldots, v_n\}$. On considère l'algorithme `App` suivant :

— Soit $A' = \{(v_i, v_j) \in A : i < j\}$ (A' contient les arcs (v_i, v_j) de A avec $i < j$), et $A'' = \{(v_i, v_j) \in A : i > j\}$.
— Si $|A'| \geq |A''|$ renvoyer A', sinon renvoyer A''.

1. A quoi sont égaux A' et A'' dans le graphe de la figure 1 (partie gauche) ?
2. Expliquer pourquoi `App` renvoie une solution réalisable pour le problème SGSS (quel que soit le graphe).
3. Montrer que `App` est 1/2-approché pour le problème SGSS.

4. Donner un graphe (très simple !) où la taille d'une solution optimale est exactement égale à deux fois la taille de la solution renvoyée par `App`.

> 1. $A' = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_4), (v_3, v_4)\}$ et $A'' = \{(v_4, v_1)\}$.
> 2. A' contient uniquement des arcs (v_i, v_j) avec $i < j$. La suite des indices des sommets dans tout chemin est donc strictement croissante, on ne peut avoir de circuit. A' est une solution réalisable. De même, dans (V, A'') la suite des indices des sommets dans tout chemin est strictement décroissante, on ne peut avoir de circuit. A'' est réalisable.
>
>   `App` renvoie donc une solution réalisable.
> 3. $A' \cup A'' = A$, donc le plus grand des deux ensembles est de taille au moins $|A|/2$. La valeur optimale est au plus $|A|$, donc la taille de la solution renvoyée est au moins la moitié de la valeur optimale.
> 4. On prend trois sommets v_1, v_2, v_3 et deux arcs (v_2, v_1) et (v_2, v_3). $|A'| = |A''| = 1$ alors que le graphe n'a pas de circuit (donc la valeur optimale est 2).

Question 3 (1/14) — On s'intéresse maintenant à la complexité du problème. Montrer que le problème SGSS-D est dans NP.

> Certificat = ensemble d'arcs. On peut vérifier en temps polynomial qu'un graphe est sans circuit. Il suffit par exemple de voir s'il y a un sommet sans prédécesseur : s'il n'y en a pas le graphe a un circuit. S'il y en a, on le(s) supprime, et on itère. Si on a tout supprimé le graphe est sans circuit. (ou ordre topologique, ou parcours avec arc arrière,...).
> NDLR : On peut être peu exigeant sur la justification du test en temps polynomial.

Question 4 (5/14) — On rappelle que dans un graphe *non orienté* $G = (V, E)$, un stable V' est un ensemble de sommets deux-à-deux non adjacents (chaque arête de E a au plus une extrémité dans V'). On rappelle également que le problème STABLE-D suivant est NP-complet :

— Entrée : un graphe non orienté G et un entier k.

— Question : existe-t-il dans G un stable de taille au moins k ?

Nous allons faire une réduction de STABLE-D à SGSS-D. Soit $G = (V, E)$ un graphe non orienté. On définit le graphe orienté $R = f(G)$ suivant (voir la figure 2 pour une illustration) :

— Pour chaque sommet v_i de V on considère deux sommets b_i et c_i dans R, avec l'arc (b_i, c_i).

— Pour chaque arête (v_i, v_j) de G, on met les deux arcs (c_i, b_j) et (c_j, b_i).

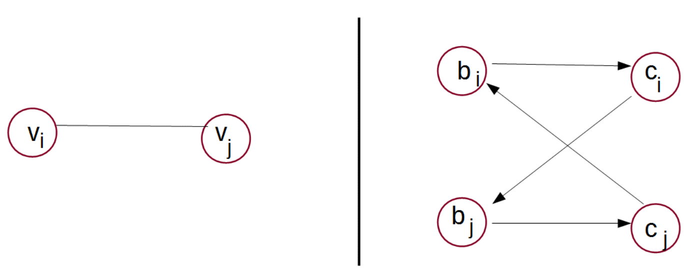

FIGURE 2 – Illustration de la construction sur deux sommets v_i et v_j adjacents.

On notera A_0 l'ensemble des arcs créés à l'étape 1 (les arcs (b_i, c_i)) et A_1 l'ensemble des arcs créés à l'étape 2.

Soit $B = \{b_1, \ldots, b_n\}$, et $C = \{c_1, \ldots, c_n\}$. On remarquera que dans R tout arc a une extrémité dans B et l'autre dans C. Ainsi, tout circuit de R alterne les sommets de B et de C. On remarquera aussi que b_i a un unique successeur, le sommet c_i, et que c_i un unique prédécesseur, le sommet b_i.

1. Si l'on note n le nombre de sommets de m le nombre d'arêtes de G, combien R a-t-il de sommets et d'arcs ?
2. Soit S un stable de G de taille k. On considère l'ensemble d'arcs A' contenant (1) tous les arcs de A_1, et (2) l'arc (b_i, c_i) si et seulement si v_i est dans S.
   (a) Quelle est la taille k' de A' ?
   (b) Montrer que A' est un ensemble d'arcs sans circuit de R. On pourra raisonner par l'absurde, en supposant l'existence d'un circuit C, considérer deux sommets consécutifs c_j, b_i de C, et s'intéresser aux sommets v_i et v_j dans G.
3. Réciproquement, soit A' un ensemble d'arcs sans circuit de R de taille k'. On suppose que A' contient A_1 (on admet que cette condition n'est pas restrictive). Considérons l'ensemble S contenant l'ensemble des sommets v_i tels que $(b_i, c_i) \in A'$. Montrer que S est stable. Quelle est sa taille ?
4. Conclure.

1. R a $2n$ sommets (deux sommets b_i et c_i par sommet v_i, et $n+2m$ arcs (un arc par sommet v_i dans A_0, et deux arcs par arête dans A_1).
2. A' est de taille $k' = 2m + k$.

   Supposons qu'il y ait un circuit C dans R restreint aux arcs de A'. Soit c_j un sommet de C (C ne peut pas passer que par des sommets B), et b_i le successeur de c_j dans C. L'arête (v_i, v_j) est donc dans G. b_i a un unique successeur, c_i, donc C passe par (b_i, c_i). De même, c_j a un unique prédécesseur dans R, le sommet b_j. Donc C passe par (b_j, c_j). Ainsi, $v_i \in S$ et $v_j \in S$. S étant stable, (v_i, v_j) n'est pas une arête de G, contradiction.
3. Soit A' un ensemble d'arête sans circuit de taille $k' = 2m + k$ contenant A_1. A' contient donc $k' - 2m = k$ arcs dans A_0. Considérons S l'ensemble des sommets v_i tel que l'arc (b_i, c_i) est dans A'. S'il existait une arête (v_i, v_j) avec v_i et v_j dans S, alors A' contiendrait les arcs (b_i, c_i) et (b_j, c_j) (car b_i et c_j sont dans S), (c_i, b_j) et (c_j, b_i) (car A' contient A_1), ce qui forme un circuit.
4. G a un stable de taille k si et seulement si $R = f(G)$ a un ensemble d'arc sans circuit de taille $2m + k$. On a une réduction polynomiale de STABLE-D à SGSS-D. STABLE-D étant NP-complet, et SGSS-D étant dans NP, SGSS-D est NP-complet.

Question 5 (2/14) — Que pensez-vous du problème SGSS dans un graphe non orienté ? On cherche donc à trouver un ensemble d'arêtes de taille maximale tel que le graphe réduit à ces arêtes ne contienne pas de cycle. On pourra supposer dans un premier temps que le graphe est connexe.

Dans un graphe connexe, tout graphe sans cycle a au plus $n-1$ arêtes, et on peut trouver une solution avec $n-1$ arêtes en prenant n'importe quel arbre couvrant. Plus généralement, on peut trouver une solution optimale en prenant un arbre couvrant dans chaque composante connexe (de taille $n-t$ où t est le nombre de composantes connexes).