

Toutes vos réponses doivent être justifiées. La langue pour vos réponses peut-être le français ou l'anglais, mais pas les deux! Documents (cours et notes personnelles sur papier) autorisés. Les notes de cours officielles seront disponibles pendant l'épreuve.

VERTEX COVER

Question 1. Rappelez la définition du problème VERTEX COVER de taille minimum.

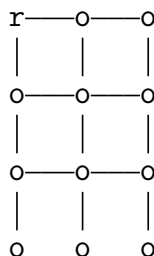
SOLUTION. [3 pts | $\sum 3.0$] L'entrée est un graphe G . La sortie est une "couverture par sommet" de G , c'est-à-dire un ensemble de sommets $S \subseteq V(G)$ telle que chaque arête de G possède au moins une extrémité dans S . Il s'agit ici de déterminer la plus petite couverture par sommet pour G .

On dit qu'un sommet u d'un arbre enraciné T est *interne* si u n'est pas une feuille. On considère l'algorithme suivant, qui suppose que le graphe G est connexe.

Algorithme ApproxVC(G)

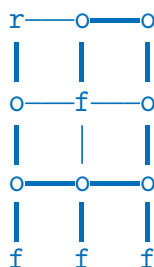
1. Calculer pour G un arbre couvrant T en profondeur d'abord.
2. Renvoyer l'ensemble S des sommets internes de T .

Question 2. Appliquez ApproxVC au graphe G_0 ci-dessous de sorte que l'arbre T ait pour racine r . Donnez la taille de l'ensemble S renvoyé. Précisez T et S sur un dessin de G_0 .



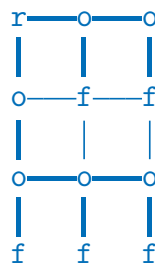
SOLUTION. [3 pts | $\sum 6.0$] On peut prendre par exemple les arêtes épaïses comme ci-dessous, qui balaye des 9 premiers sommets en tournant, laissant 4 feuilles marquées d'un "f". L'ensemble S est donc formé des 12 sommets de G_0 moins ces 4 feuilles, soit $|S| = 8$.

D'autres arbres sont possibles, comme un chemin zigzagant les 9 premiers sommets de la grille de gauche à droite ou de bas en haut, laissant seulement 3 feuilles, les trois sommets de degré 1. Dans ce cas, on trouvait $|S| = 9$.



Attention! Il était important que l'arbre soit obtenu à partir d'un parcours en profondeur, et donc tous les arbres couvrant de racine r n'étaient pas possibles.

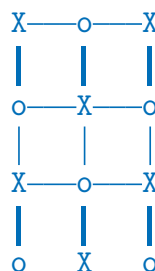
Par exemple, l'arbre couvrant de racine r ci-dessous n'est pas possible.



On observe d'ailleurs que l'arête "f—f" n'aurait pas été couverte par les sommets internes d'un tel arbre T .

Question 3. Trouvez un VERTEX COVER de taille minimum pour le graphe G_0 (un dessin suffit). Justifiez la minimalité.

SOLUTION. [3 pts | $\sum 9.0$] L'ensemble représenté par des "X" est une couverture par sommets de taille 6. C'est optimal car G_0 possède un couplage de 6 arêtes (donc parfait puisqu'il y a 12 sommets), celles verticales épaisses. La solution d'ailleurs consiste à sélectionner une extrémité sur deux de ce couplage.



Question 4. Démontrez que, pour les graphes connexes, l'algorithme ApproxVC est une 2-approximation pour VERTEX COVER de taille minimum. [Aide : Coloriez les sommets de S .]

SOLUTION. [6 pts | $\sum 15.0$] Comme vu dans le cours, il y a trois points à démontrer.

1. L'algorithme est polynomial. C'est évident, l'algorithme se résume à un parcours en profondeur d'abord (DFS) qui est linéaire en $|V(G)| + |E(G)|$.
2. L'algorithme calcule bien une solution valide, soit une couverture par sommet. En effet, les seules arêtes qui n'ont aucune extrémité dans S doivent connecter deux feuilles de T . C'est impossible dans un DFS.
3. Le facteur d'approximation est 2. C'est le point le plus délicat. L'idée est de montrer que G , possède un couplage M de taille $|M| \geq |S|/2$, où S est l'ensemble renvoyé par ApproxVC(G). Cela suffit, puisque la solution optimale, disons OPT(G), doit alors vérifier $|\text{OPT}(G)| \geq |M| \geq |S|/2$, ce qui revient à dire que $|\text{ApproxVC}(G)| = |S| \leq 2 \cdot \text{OPT}(G)|$, soit un facteur d'approximation de 2.

Montrons donc que G possède un couplage M de taille au moins $|S|/2$. On va se contenter de montrer que T possède un tel couplage. On colorie les sommets de S en deux couleurs. Pour cela il suffit de colorier d'une même couleur tous ceux à distance paire de r , et de l'autre couleur ceux à distance impaire. Notons W l'ensemble de tous les sommets de la couleur la plus représentée. Clairement, $|W| \geq |S|/2$ puisque les deux couleurs partitionnent les sommets de S .

On observe que, pour tout $w \in W$, on a : (1) w possède au moins un enfant dans T , disons $f(w)$, car $W \subseteq S$ n'est composé d'aucune feuille ; et (2) $f(w) \notin W$, car sinon on aurait deux sommets voisins dans T de la même couleur.

Il suit de cette double observation que l'ensemble d'arêtes $\{w - f(w) : w \in W\}$ forment un couplage M de T , et sa taille est $|M| = |W| \geq |S|/2$ comme requis.

On a donc montré que ApproxVC est une 2-approximation.

Question 5. Est-ce que l'algorithme ApproxVC est encore correct si au lieu de renvoyer les sommets internes de T on renvoie tous les sommets de T de degré > 1 ? Justifiez.

SOLUTION. [3 pts |Σ18.0] Considérer un graphe composé d'une seule arête était un possible exemple montrant que c'est faux, car dans cet exemple aucun sommet n'est de degré >1 (et donc l'ensemble vide sera renvoyé) alors qu'il faut au moins un sommet.

Mais c'est aussi faux de manière plus générale. Remarquons que dans un DFS d'un graphe G , autant il n'est pas possible d'avoir une arête entre deux feuilles, autant il est tout à fait possible d'avoir une arête entre deux sommets qui sont de degré 1 dans l'arbre (et donc qui ne sera pas couverte par l'algorithme). Un exemple concret consiste à prendre pour G un cycle de longueur 3.

De manière générique, cela peut arriver dans un graphe G obtenu en choisissant un arbre arbitraire T_0 de racine r_0 , puis en ajoutant un nouveau sommet r connecté à r_0 et f , où f est une feuille quelconque de T_0 . Donc G est un cycle si T_0 est un chemin. Supposons que la première arête du parcours DFS depuis r dans G choisisse $r - r_0$. Dans ce cas, l'arbre calculé par $\text{ApproxVC}(G)$ est $T = \{r - r_0\} \cup T_0$, et comme $\deg_T(r) = \deg_T(f) = 1$, l'arête $r - f$ de G ne sera pas couverte.

Question 6. *Donnez une instance critique pour ApproxVC. [Attention ! L'instance critique ne doit pas être de taille constante.]*

SOLUTION. [3 pts |Σ21.0] Un chemin de longueur L paire où la racine de T est l'une des extrémités. T est le chemin lui-même. La solution optimale est de taille $L/2$ en prenant un sommet sur deux, alors qu'ApproxVC va renvoyer $n - 1 = L$ sommets, soit exactement le double de l'optimal.

Question 7. *Rappelez brièvement le principe de l'algorithme Glouton pour SET COVER mais appliqué à VERTEX COVER.*

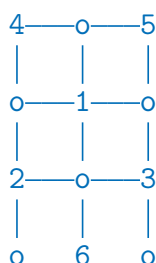
SOLUTION. [3 pts |Σ24.0] Pour résoudre VERTEX COVER grâce à l'algorithme Glouton de SET COVER d'entrée $(B, \mathcal{F})$, il faut prendre $B = E(G)$ et pour $\mathcal{F} = \{E_u : u \in V(G)\}$ où E_u est l'ensemble des arêtes incidentes à u dans G . L'algorithme Glouton consiste à sélectionner itérativement le sommet u couvrant le plus d'arêtes restant à couvrir.

Question 8. *En fonction d'un paramètre pertinent de G_0 , quel facteur d'approximation théorique devrait atteindre l'algorithme Glouton sur G_0 .*

SOLUTION. [3 pts |Σ27.0] Le facteur d'approximation est au plus la série harmonique $H(k)$ où k est la taille maximum d'un ensemble de $\mathcal{F}$ pris dans la solution, ici un ensemble d'arêtes incidentes à un sommet. Le paramètre pertinent est donc le degré maximum du graphe, soit $k = 4$ pour G_0 , d'où un facteur d'approximation théorique d'au plus $H(4) = 1 + 1/2 + 1/3 + 1/4 \approx 2.08$.

Question 9. *Appliquez cet algorithme Glouton sur G_0 . Quelle est la taille de la couverture renvoyée ? Dessinez cette solution.*

SOLUTION. [3 pts |Σ30.0] L'ordre de sélection des sommets pouvait être le suivant (c'est celui qui donnait la couverture la plus petit avec cet algorithme) :



La taille de la couverture renvoyée est donc 6, soit une solution optimale. Pour le sommet 2, on pouvait choisir le voisin bas de 1, ce qui menait à une couverture de taille 7. En choisissant l'autre sommet de degré 4 en premier, on pouvait obtenir 6 ou 7 selon le deuxième sommet (voisin ou pas du premier).

DOMINATING SET

Question 10. *Rappelez la définition du problème DOMINATING SET de taille minimum.*

SOLUTION. [3 pts | Σ 33.0] L'entrée est un graphe G . La sortie est un "ensemble dominant", soit un ensemble $S \subseteq V(G)$ tel que tout sommet de G qui n'est pas dans S ait un voisin dans S . Il s'agit ici de déterminer le plus petit ensemble dominant pour G .

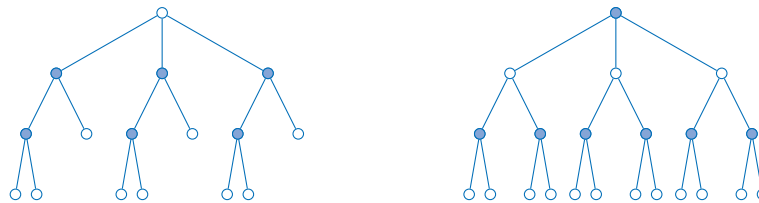
Question 11. *Donnez le facteur d'approximation théorique de l'algorithme Glouton appliqué à DOMINATING SET pour un arbre binaire à n sommets.*

SOLUTION. [3 pts | Σ 36.0] Le facteur d'approximation théorique de l'algorithme Glouton est $H(4) \approx 2.08$ puisque que les ensembles couvrant pris dans $\mathcal{F}$ (qui sont des boules de rayon 1) contiennent $k \leq 4$ sommets dans le cas d'un arbre binaire : le sommet, ses deux enfants, et son parent.

Question 12. *Donnez un algorithme polynomial résolvant DOMINATING SET de taille minimum pour une forêt exprimé sous la forme de règles de réduction. Justifiez.*

SOLUTION. [6 pts | Σ 42.0] Tout comme VERTEX COVER, il existe un ensemble dominant de taille optimale contenant le parent d'une feuille et pas la feuille. Il est donc valide de sélectionner ce parent, de le supprimer, et de marquer ses voisins comme dominés. À noter que pour que la notion de parent ait un sens, il est nécessaire de choisir au départ un sommet racine dans chaque composante connexe de la forêt. Ne pas le faire poserait problème pour les composantes avec une seule arête par exemple.

Il faut maintenant gérer correctement la version annotée du problème puisqu'avec cette règle certains sommets deviennent marqués. Car, comme on l'a déjà vu, il est assez naïf de supprimer le sommet sélectionné et tous ses voisins, ou alors aucun. On voit que ces stratégies naïves échouent pour au moins l'un des deux arbres ci-dessous, où leurs ensembles dominants minimum sont uniques et les suivants :



On remarque aussi que les sommets de degré ≤ 1 marqués (feuille ou sommet isolé) peuvent-être supprimés, car il est facile de vérifier qu'il existe un ensemble dominant minimum (pour la version annotée) ne contenant pas ces sommets. Bien évidemment, les sommets isolés non-marqués doivent être sélectionnés.

En résumé :

- (R0) Sélectionner un sommet isolé non-marqué.
- (R1) Supprimer un sommet marqué de degré ≤ 1 .
- (R2) Sélectionner le parent d'une feuille non-marquée, marquer ses voisins, et le supprimer.

On en déduit l'algorithme qui consiste à appliquer autant que possible (R0), (R1) ou (R2), l'ordre n'ayant pas d'importance, jusqu'à avoir marqué ou supprimé tous les sommets.

Cet algorithme est polynomial, puisqu'on applique (R0), (R1) ou (R2) au plus n fois, n étant le nombre de sommets de la forêt, chacune d'elles se testant en temps $O(n)$.