

## ANALYSE D'ALGORITHME (projet de programmation)

---

*Ce devoir est à rendre avant le lundi 6 janvier 2010 midi. Le contrôle continu est **obligatoire** et compte 50% de la note finale du module (qui est la moitié de l'UE). Vous devrez rendre **deux** fichiers impérativement par courriel à gavoille@labri.fr. Fichier 1 : le **source** de votre programme (c ou c++ compilable sous Linux). Fichier 2 : le **rapport** au format .pdf ou .ps.*

### Le problème

Il s'agit de programmer un algorithme permettant de calculer une **couverture par sommets de taille minimum** dans des graphes de largeur arborescente au plus deux. En plus de la taille, votre programme devra fournir une couverture possible.

Pour vous pourrez vous inspirer de l'algorithme vu en cours pour le problème de la 3-coloration lorsqu'une décomposition arborescente du graphe (de largeur bornée) est donnée.

### Les graphes

Pour simplifier, on s'intéressera uniquement aux graphes *planaire-extérieurs*. Ces graphes possèdent une représentation planaire (dessin sur le plan sans croisement d'arêtes) de sorte que tous les sommets sont sur le bord de la face extérieure. Par exemple, les arbres et les cycles sont des graphes planaire-extérieurs. On peut montrer que les graphes planaire-extérieurs sont de largeur arborescente au plus deux.

### Décomposition arborescente

Il est facile de calculer une décomposition arborescente optimale pour ces graphes en utilisant la propriété suivante : si  $G$  est un graphe planaire-extérieur alors son degré minimum est au plus deux. De plus le graphe  $G'$  obtenu en supprimant un sommet  $x$  de

degré minimum et en ajoutant une arête  $e$  entre les deux voisins de  $x$  est encore un graphe planaire-extérieur. (Si  $\deg(x) < 2$ , on n'ajoute pas d'arête.)

Dans toute décomposition arborescente  $T'$  du sous-graphe  $G'$  décrit précédemment, il existe un sac  $X$  contenant les voisins de  $x$  (puisque ces voisins sont connectés dans  $G'$  par une nouvelle arête  $e$ ). Donc pour obtenir une décomposition arborescente  $T$  pour  $G$  il suffit de : 1) calculer (récursivement) une décomposition  $T'$  pour  $G'$  ; 2) déterminer un sac  $X'$  de  $T'$  contenant les voisins de  $x$  ; 3) former un sac  $X$  contenant  $x$  et ses voisins ; et 4) ajouter une arête entre  $X$  et  $X'$  pour former le nouvel arbre  $T$ .

## Quelques consignes

1. Pour tester votre programme, vous utiliserez le générateur de graphe `gengraph` dont le source se trouve sur : <http://dept-info.labri.fr/~gavoille/gengraph.c>. Vous obtiendrez une aide avec `gengraph -help`. Pour créer un graphe planaire-extérieur  $G$  aléatoire de 100 sommets par exemple, lancer `gengraph outerplanar 100 -format list > G.txt`. Le générateur permet de générer des arbres aléatoires, mais aussi des chemins, des cycles, *etc.* qui sont des graphes planaires-extérieurs. Vous pouvez aussi tester `gengraph path 100 -dele 0.3 -format list` qui supprime aléatoirement avec probabilité 0.3 les arêtes d'un chemin de 100 sommets.
2. Il est inutile de prévoir une interface graphique. L'efficacité doit primer ! Votre programme devrait plutôt se présenter sous forme d'une ligne de commande et prendre comme paramètre un graphe donné sous la forme d'un fichier texte dans le format de liste d'adjacence fournit par `gengraph -format list ....`
3. Le résultat de votre programme devra donc être une liste de sommets décrivant la couverture de taille minimum. Je vous conseil d'écrire une routine permettant de vérifier que la solution est bien correcte. Pensez que pour les chemins ou les cycles la solution optimale est facile à déterminer à la main.
4. Dans votre rapport vous donnerez l'algorithme et analyserez sa complexité totale (chargement du graphe en mémoire, décomposition, et recherche de la solution). Votre rapport devrait comporter entre 5 et 10 pages maximum.
5. Vous pouvez présenter des améliorations qui vous semble améliorer la vitesse de votre programme. Indiquer jusqu'à combien de sommets vous pouvez résoudre le problème.