Université de Bordeaux
Année 2010-2011

Master2 Informatique, spécialité AMF
C. Gavoille

Analyse d'Algorithme
(projet de programmation)

---

*Ce devoir, qui peut être réalisé par **binôme**, est à rendre avant le mardi **21 décembre 2010** midi. Ce contrôle continu est **obligatoire** et compte pour 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++ standard compilable sous Linux). Fichier 2 : **le rapport** au format .pdf ou .ps. Je ne veux pas de d'archives et de fichiers compressés ! Grâce au générateur de graphes, je n'ai, en principe, pas besoin des jeux d'essais.*

Le problème

Il s'agit de programmer un algorithme permettant de calculer un **ensemble dominant** de taille k (voir de taille minimum) pour les graphes planaires, mais aussi les graphes de degré maximum $\leqslant d$. En plus de la taille optimale, votre programme devra fournir une solution possible, c'est-à-dire un ensemble dominant pour le graphe.

Pour vous pourrez vous inspirer de l'algorithme à base d'arbre borné de recherche vu en cours et de complexité $O(8^k \cdot n)$. Votre programme devra également donner le nombre de noeuds explorés jusqu'à la solution.

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`.
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 un des formats de `gengraph`.
3. Dans votre rapport vous donnerez l'algorithme que vous avez programmé et vous 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.

4. Vous pouvez présenter des améliorations qui vous semble améliorer la vitesse de votre programme. La partie expérimentation devra être une partie importante de votre rapport. Indiquer les temps de calcul et jusqu'à combien de sommets (ou jusqu'à quelle valeur de k ou de d) vous pouvez résoudre le problème.
5. Pour finir, voici quelques façons de générer un graphe planaire avec `gengraph` (ici `n` est le nombre de sommets souhaités du graphe)
   - `gengraph path n`
   - `gengraph tree n`
   - `gengraph outerplanar n`
   - `gengraph kpage n 2`
   - `gengraph gabriel n`
   - `gengraph td-delaunay n`
   - `gengraph rng n`
   - ...

   Il y en a d'autres : `path`, `cycle`, `mesh`, `house`, `hajos`, `sierpinski`, `hanoi` ... Pour générer des graphes d-dégénérés, essayez les graphes comme `kpage`, `kout`, `arboricity`, `tw`, `pw`, ... Si vous avez `dot` (Graph Viz) installé sur votre système, vous pouvez visualiser ces graphes avec l'option `-visu`.