
Le projet est à rendre par courriel (gavoille@labri.fr) au plus tard ***jeudi 27 avril*** *2006 minuit. Vous devez m'envoyer vos fichiers sources java (1 par exercice) que je testerai sous ViSiDiA (en synchrone), ainsi qu'un rapport sous forme* ***lisible*** *(.ps, .pdf, ou .txt). Si vous pensez que cet util, vous pouvez également m'envoyer un fichier de graphe d'exemple sur lequel votre algorithme fonctionne.*

Le rapport : il devra contenir au plus ***3 pages****. Il décrira le plus clairement possible les algorithmes utilisés et vous donnerez pour chacun d'eux la complexité MESSAGE et TEMPS, avec la notation O(). Vous veillerez à ce que les algorithmes et les complexités donnés dans votre rapport correspondent à ceux de vos programmes. Il n'y a pas une, mais plusieurs réponses possibles. La notation tiendra compte de la correspondance entre le rapport et votre programme.*

Excercice 1: taux de croissance

Soit G un graphe. Pour tout sommet u de G et tout entier $r \geqslant 0$, on note $B(u,r)$ l'ensemble des sommets à distance au plus r de u, c'est-à-dire la boule de rayon r centrée en u. Notez que $B(u,0) = \{u\}$. Le *taux de croissance* du sommet u est le plus petit entier c tel que pour tout $r \geqslant 0$, $|B(u,r)|/|B(u,r/2)| \leqslant c$. (Rappel: si B est un ensemble, $|B|$ est sa cardinalité)

Écrire un programme en ViSiDiA en mode synchrone qui calcul le taux des sommets d'un graphe. À la fin de l'exécution, chaque sommet devra contenir dans son étiquette "label" son taux de croissance.

Excercice 2: dimension doublante

Soit G un graphe. La *dimension doublante* de G est le plus petit entier k tel pour chaque sommet u de G et chaque entier $r \geqslant 0$, il existe un ensemble S d'au plus k sommets de G tels que pour tout sommet v de $B(u,r)$ il existe un sommet $w \in S$ tel que $d(v,w) \leqslant r/2$. (On rappelle que $dv,w)$ est la distance dans le graphe entre v et w). Dit autrement, la boule centrée en u de rayon r est couverte par (ou contenue dans) au plus k boules de rayon deux fois moins.

Calculer la dimension doublante est NP-complet. On propose alors d'approximer la dimension doublante par la "méthode des r-nets". Un r-net S est un sous-ensemble de sommets de G maximal pour la propriété suivante : deux sommets distincts de S sont à une distance $> r$ (c'est-à-dire que si l'on ajoute un seul sommet à S alors il perd cette propriété). Un ensemble maximal indépendent (MIS) est exactement un 1-net. Le calcul d'un r-net peut se faire par un algorithme glouton (séquentiel) suivant :

```
U := V(G); M := 0
Tant que U<>0 faire:
1. choisir un sommet u dans U
2. M := M+{u}
3. U := U-B(u,r)
Retourner M
```

Pour approximer la dimension doublante, on calcule simplement un $r/2$-net S, pour tous les $r \geqslant 0$, puis on évalue pour chaque sommet u la quantité $|B(u,r) \cap S|$, et on prend la valeur maximum atteinte pour un sommet. Donc l'algorithme séquentiel d'approximation de la dimension doublante est le suivant :

```
k := 0
Pour tout r de 0 à n faire:
1. Calculer un r/2-net S
2. Pour tout u de V(G) faire: k := max{k,|B(u,r) INTER S|}
Retourner k
```

Écrire un programme en ViSiDiA en mode synchrone qui calcul une approximation de la dimension doublante d'un graphe par la méthode des r-net. A la fin de l'exécution, tous les sommets devront contenir cette valeur commune dans une de leur étiquette ("label" par exemple). Si vous avez besoin de distinguer un sommet, vous utiliserez le sommet 0. Dans la construction, n'hesiter pas à colorier les sommets d'un $r/2$-net.

Rappels :

`putProperty("label", String)` permet de changer l'étiquette du sommet, et donc d'afficher n'importe quel résultat en modifiant `String`. Par exemple, vous pourrez afficher le nombre de sommets à une certaine distance.

`putProperty("toto", String)` permet de définir une nouvelle étiquette, `toto`, pour un sommet qui sera visualisable en sélectionnant le sommet, puis en cliquant sur l'icône information "I" en bleu du simulateur.