

TP1 – Algorithmes gloutons

Le but de ce TP est d'étudier le problème de l'indépendant maximum. On rappelle qu'un *ensemble indépendant* est un ensemble de sommets deux à deux non reliés dans un graphe. L'objectif est, étant donné un graphe, de trouver un ensemble indépendant aussi grand que possible.

1 Prise en main du programme

Téléchargez l'archive <https://perso.liris.cnrs.fr/tpierron/RO2023/TP1-etu.zip>. Elle contient :

- Une bibliothèque de gestion des graphes `graph.cpp` `graph.hpp`.
- Un `Makefile`, permettant de compiler le programme en tapant la commande `make` dans un terminal.
- Un sous-dossier `data/` contenant des graphes (un par fichier) au format suivant :
 - la première ligne du fichier contient le nombre n de sommets dans le graphe,
 - toutes les autres lignes du fichier contiennent une arête du graphe sous la forme de deux entiers $u v$ séparés par un espace, ou u et v sont les identifiants des deux sommets (distincts) qui sont les extrémités de l'arête,
 - l'identifiant d'un sommet est un entier compris entre 0 et $n - 1$.
- Un fichier à compléter `main.cpp`.

Le seul fichier que vous avez à modifier est `main.cpp`. Il n'est pas nécessaire de changer les autres.

Question 1. Ouvrez le fichier `graph.hpp` et `main.cpp`. Compiler puis exécuter le programme. Que font les méthodes `print` et `show` ?

Question 2. Écrire une fonction `clique` qui prend en paramètre un entier n et génère une clique à n sommets.

Question 3. Écrire une fonction `biclique` qui prend en paramètre deux entiers n et p et génère un graphe à $n + p$ sommets répartis en deux ensembles de taille n et p et contenant toutes les arêtes entre les sommets de $[0, n - 1]$ et ceux de $[n, n + p - 1]$.

2 Algorithme glouton

On souhaite maintenant implémenter l'algorithme glouton (du TD) consistant à construire un ensemble indépendant I maximal par inclusion itérativement, en parcourant les sommets et en ajoutant un sommet à I s'il est adjacent à aucun sommet déjà présent dans I . On rappelle qu'un ensemble indépendant maximal par inclusion est un ensemble indépendant X tel que tout sommet $v \in V$ est soit dans X soit adjacent à un sommet de X .

Question 4. Écrire une fonction `greedy_det` qui implémente l'algorithme glouton ci-dessus. Quelle est la complexité de cet algorithme ?

Question 5. Tester votre algorithme sur les graphes de `data/`. Jusqu'à quelle taille de graphes pouvez-vous répondre en moins d'une minute ?

3 Résolution exacte

Afin de comparer nos solutions gloutonnes aux solutions optimales, on a besoin d'une fonction qui résout le problème de manière exacte, c'est-à-dire qui détermine toujours un ensemble indépendant de taille maximale. Dans cet objectif, on va générer tous les ensembles indépendants de G et conserver le plus grand qu'on aura rencontré.

La génération des ensembles indépendants repose sur la propriété suivante :

Soit I un ensemble indépendant d'un graphe G , et v un sommet de G . On a deux choix :

- Soit $v \notin I$, et I est un ensemble indépendant de $G \setminus v$.
- Soit $v \in I$, et $I \setminus \{v\}$ est un ensemble indépendant du graphe G_v obtenu à partir de G en enlevant v et ses voisins.

Ainsi, si on essaye de construire un ensemble indépendant I dans G , on peut choisir un sommet v dans G , et chercher récursivement les ensembles indépendants de $G \setminus v$ ou de G_v . On a donc besoin d'une fonction qui peut calculer un ensemble indépendant de G qui évite certains sommets.

Question 6. Compléter la fonction `exact_aux` de `main.cpp`. Cette fonction prend en entrée un graphe G , un vecteur `is_free` de $|V(G)|$ booléens et un entier i . Elle renvoie un plus grand ensemble indépendant I de G tel que :

- I contient uniquement des sommets j tels que `is_free[j]` est vrai.
- Si $j < i$ et `is_free[j]` est vrai, alors $j \in I$.

Question 7. À l'aide de la fonction précédente, compléter `exact`, qui prend en entrée un graphe et renvoie un de ses ensembles indépendants maximum.

Question 8. Tester votre algorithme sur les graphes de `data/`. Jusqu'à quelle taille de graphes pouvez-vous répondre en moins d'une minute ?