

Dans ce projet, vous devrez concevoir différents programmes concernant notamment la notion de « *sous-graphe désert* ». Un *sous-graphe désert* dans un graphe non-orienté $G = (S, A)$, est un sous-ensemble X de S tel que pour couple de sommets $x, y \in X$, l'arête ne figure pas dans G , c'est-à-dire que $\{x, y\} \notin A$. En d'autres termes, un *sous-graphe désert* est un sous-ensemble de sommets non connectés entre eux.

Dans un premier temps, vous devrez concevoir un programme qui permet la recherche de sous-graphes déserts dans un graphe. En particulier, vous devrez concevoir des fonctions C qui permettent de résoudre les problèmes suivants :

Vérification. Concevez une fonction qui prend en entrées un graphe non-orienté G et un ensemble de sommets X , et qui vérifie si X est un sous-graphe désert de G .

Maximalité. Concevez une fonction qui prend en entrées un graphe non-orienté G et un ensemble de sommets X , et qui vérifie si X est un sous-graphe désert maximal de G , c'est-à-dire qui vérifie si X n'est pas strictement inclus dans un ensemble de sommets qui serait également un sous-graphe désert de G .

Calcul de sous-graphe désert maximal. Concevez une procédure qui prend en entrée un graphe non-orienté G et qui fournit en résultat un sous-graphe désert maximal de G .

Calcul de sous-graphe désert maximum. On s'intéresse maintenant au calcul de sous-graphes déserts de taille maximum dans un graphe. Ce problème n'est pas toujours simple à résoudre en pratique, selon le graphe fourni en entrée. Aussi, vous devrez concevoir ici au moins deux types de procédures :

- *méthode exacte* : concevez une procédure qui prend en entrée un graphe non-orienté G et calcule un sous-graphe désert de taille maximum (on dit *sous-graphe désert maximum*) du graphe G .
- *méthode incomplète* : concevez une procédure qui prend en entrée un graphe non-orienté G et calcule un sous-graphe désert non nécessairement maximum, aussi proche que possible en taille, de l'optimum (i.e. la taille maximum).

Il vous faudra bien entendu concevoir l'environnement nécessaire pour tester et expérimenter vos fonctions C , incluant notamment la saisie et la lecture dans des fichiers de jeux de données.