TD 10. Recherche exhaustive rapide

Exercice 1. 3-SAT

Soit φ une formule 3-CNF et ℓ un littéral apparaissant dans φ.

1. Montrer que si $\neg\ell$ apparaît dans φ, alors il existe une clause de taille ≤ 2 dans $\varphi_{|\ell}$.
2. En déduire qu'à part pour la formule d'origine, on peut supposer dans les appels récursifs que φ a une clause de taille ≤ 2.
3. En déduire un algorithme et analyser sa complexité.

Exercice 2. *Indépendance*

Soit $G = (S,A)$ un graphe. Un *indépendant* de G est un sous-ensemble de sommets $I \subset S$ tel qu'il n'existe aucune arête entre eux : $\forall s,t \in I$, $\{s,t\} \notin A$. On cherche, dans un graphe donné, la taille du plus grand indépendant possible.

1. Quelle est la complexité d'un algorithme de recherche exhaustive pour ce problème ?

On veut exprimer l'algorithme de recherche exhaustive de manière récursive. Pour un sommet s, on note $V(s)$ l'ensemble de ses *voisins*, c'est-à-dire l'ensemble des sommets t qui ont une arête reliée à s : $V(s) = \{t \in S : \{s,t\} \in A\}$. L'algorithme récursif est alors le suivant : si G n'a aucun sommet, renvoyer 0 ; sinon soit $s \in S$ (quelconque) : soit l'indépendant maximal contient s et aucun de ses voisins (appel récursif sur $G \setminus \{s\} \cup V(s)$) soit il ne contient pas s (appel récursif sur $G \setminus \{s\}$).

2. i. Écrire l'algorithme.
   ii. Analyser sa complexité.
3. i. Que peut-on faire des sommets *isolés*, c'est-à-dire sans voisin ?
   ii. En déduire une variante de l'algorithme et analyser sa complexité.
4. Améliorer l'algorithme en examinant le cas des sommets ayant un seul voisin.