Licence Info
Algorithmique Avancée
TD 12 : 4-5/5/04

1. Ramassage
   Supposons qu'un programme C utilise des structures composées de noeuds où chaque noeud contient deux pointeurs `Gauche` et `Droite`. (Les structures ne sont pas forcément des arbres; elles peuvent avoir des sous-structures partagées et/ou des cycles.)
   Ecrire une fonction avec un paramètre de type pointeur vers un noeud qui parcourt toute la structure accessible à partir de ce noeud et applique `free()` à chaque noeud une seule fois. Il est interdit d'examiner un noeud déjà libéré.
   Vous pourriez construire une liste de pointeurs qui contient exactement un pointeur vers chaque noeud de la structure de départ. Si vous faites ça, il faudra bien sûr libérer aussi les noeuds utilisés pour votre liste.

2. Analyse
   On veut trier un tableau `T` qui contient n entiers de l'intervalle $[1 \cdots m^2]$ où m n'est pas très grand; on peut utiliser des tableaux de taille m.

   (a) Donner un algorithme qui trie le tableau selon les valeurs de $\lfloor T[i]/m \rfloor$, en temps linéaire ($O(n+m)$);

   (b) Donner un algorithme qui trie le tableau selon les valeurs de $T[i] \bmod m$, en temps linéaire ($O(n+m)$);

   (c) En utilisant les deux algorithmes précédents donner un algorithme qui trie le tableau en temps linéaire;

   (d) Comparer cet algorithme avec un algorithme qui trie le tableau en utilisant $O(n \log n)$ comparaisons;

   (e) Généraliser l'algorithme et l'analyse au tri de tableaux d'entiers positifs où le majorant est supérieur à m^2.