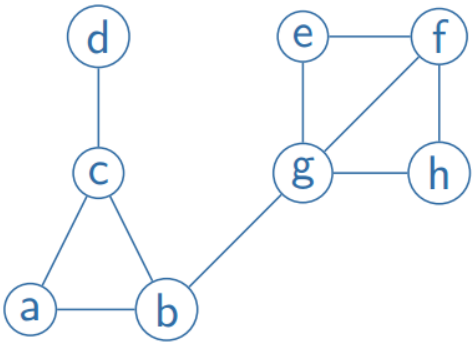
Documentation TIPE

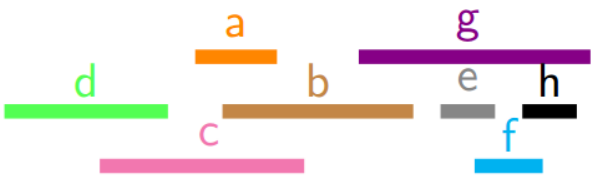
Je propose ici une implémentation en OCaml d'algorithmes de coloration de graphes d'intervalles puis de graphes cordaux.

I. Graphe d’intervalles

Un graphe est dit **d'intervalles** si on peut associer un intervalle réel à chacun de ses sommets, tel que deux intervalles ont une intersection non vide si et seulement si leurs sommets correspondants sont reliés par une arête.

Exemple de graphe d’intervalles sous forme de graphe (à droite) et sous forme d’intervalles (à gauche)

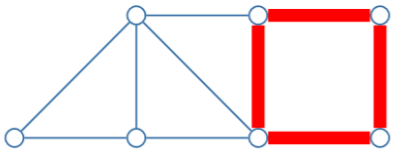




II. Graphe Cordal

Un graphe est dit **cordal** s’il ne possède aucun cycle de longueur supérieure ou égale à 4 sans corde.

Exemple de graphe *non* cordal :



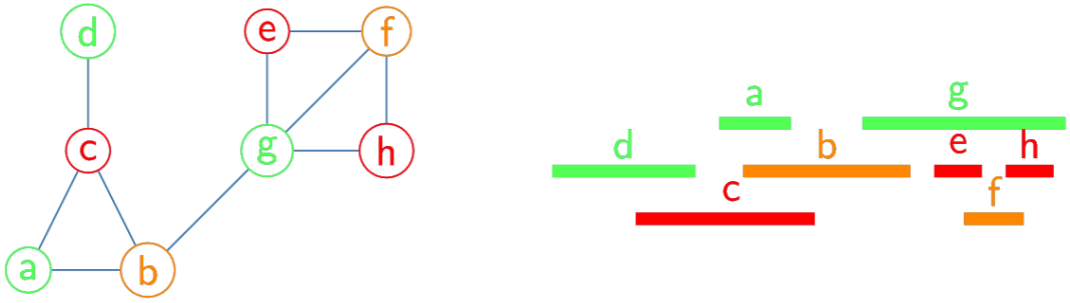
En effet le cycle en rouge est de longueur 4 mais ne possède pas de corde.

III. Coloration

Une **coloration** d'un graphe est une fonction qui attribue à chacun de ses sommets une couleur de telle sorte que deux sommets reliés par une arête soient de couleur différente.

Le **nombre chromatique** d'un graphe est le nombre minimal de couleurs nécessaires pour colorier le graphe.

Exemple de coloration du graphe d'intervalles précédent : (son nombre chromatique est 3)



Les algorithmes de coloration présentés prennent en paramètre un graphe et renvoient le couple composé du nombre chromatique et d'un tableau associant à chaque sommet une couleur. Les couleurs sont représentées par un entier à partir de 0.

Dans les deux cas, on utilise un algorithme glouton : on parcourt les sommets du graphe dans un certain ordre. Pour chaque sommet, on attribue la plus petite couleur qui n'est pas déjà prise par ses voisins (déjà coloriés).
La différence entre les algorithmes de coloration de ces deux types de graphe est l'ordre dans lequel on colorie les sommets pour obtenir une coloration optimale.

Dans le cas du graphe d'intervalle, on parcourt simplement les sommets par ordre croissant de la borne de début des intervalles. (Dans l'exemple, on commence par le sommet d, puis c, puis a,...)

Dans le cas du graphe cordal, on doit utiliser un ordre inverse d'un ordre obtenu par l'algorithme LexBFS (parcours en largeur lexicographique). On implémente donc d'abord l'algorithme LexBFS…

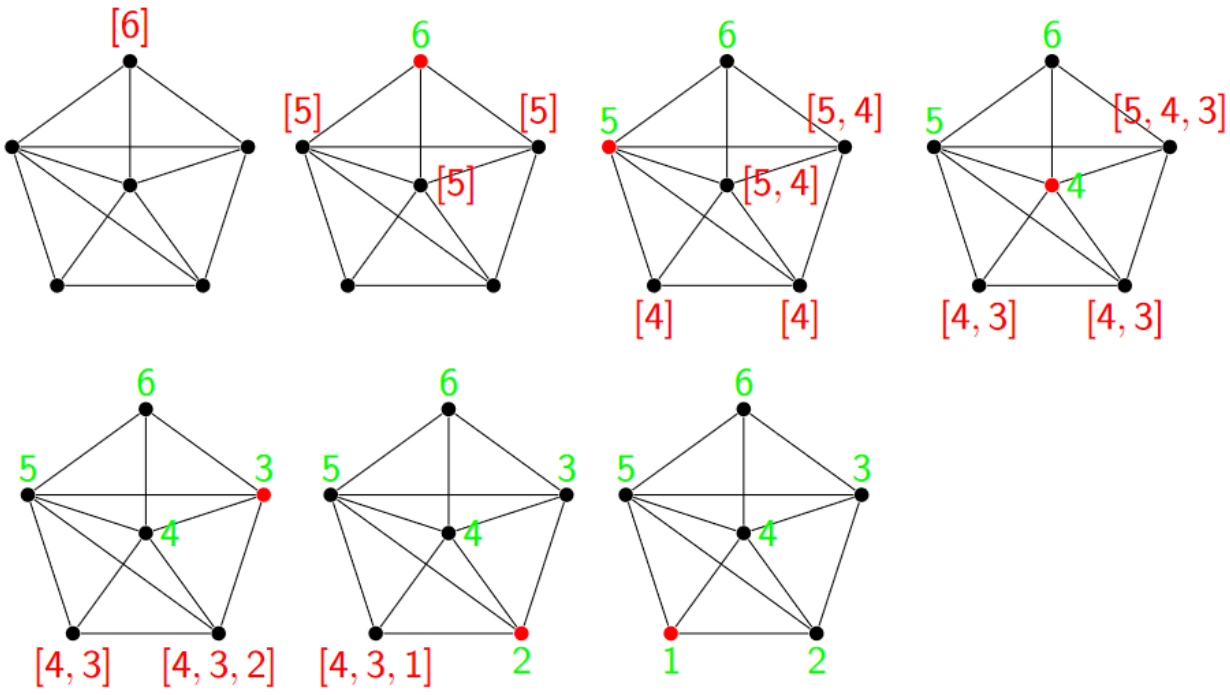
IV. LexBFS

Entrée : G= (V,E)
Sortie : σ= [x1,x2,...,x|V|] un ordre total des sommets.

Pour tout sommet x de G, label[x] ← []
label[0] ← [|V|]
Pour i de |V| à 1 :
    Choisir un sommet x d'étiquette lexicographique maximale
    σ[i] ← x
    Pour chaque voisin y de x :
        Si y n'est pas déjà dans σ :
            Concaténer i−1 à la fin de label[y]

On utilisera une structure de tas pour l'implémentation.

Exemple d'application de l'algorithme LexBFS sur un graphe cordal :



Source : Théorie des graphes, Alexandre Duret-Lutz, adl@lrde.epita.fr, 17 mars 2012