

PASQUIER PIERRE  
PRISCOGLIO FLORENT  
M1-stic

INFO 704

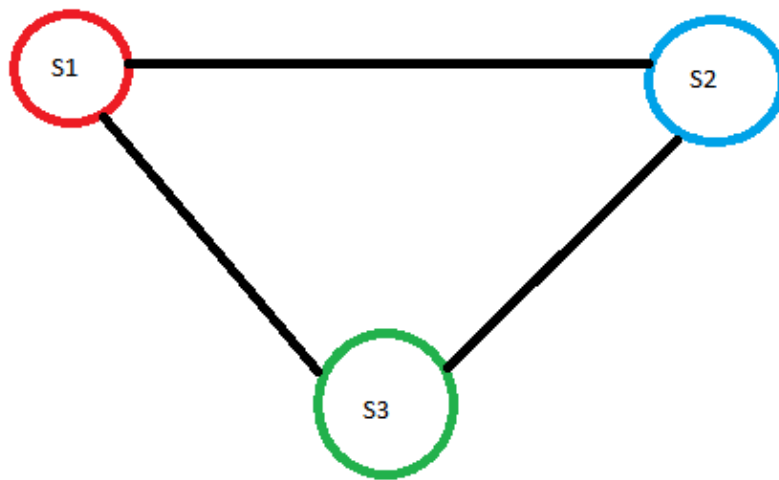
ANALYSE D'ALGORITHMES : PROBLEME DES 3 COULEURS

## Description du problème :

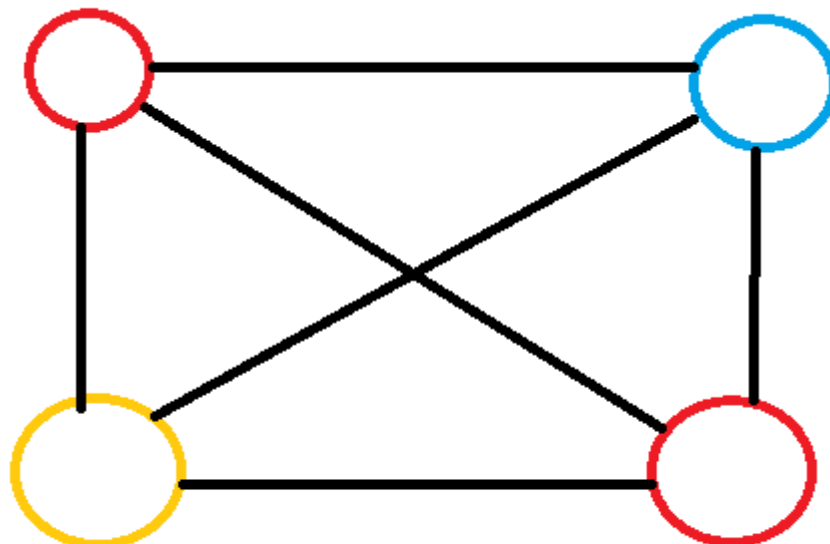
Étant donné un graphe, on veut savoir si il est possible de colorier chaque sommet de façon à ce qu'aucun sommet adjacent n'ait la même couleur.

Une solution à ce problème correspond à une assignation de couleurs.

Une solution pour un graphe à 3 sommets :



Exemple de graphe impossible à colorier avec 3 couleurs :



## Analyse de complexité :

### Version Génère et Teste :

On génère et teste toutes les instances possibles de ce problème, ce qui n'est pas du tout optimisé :

```
Fonction GeT (graphe g) :  
Début :  
    Tant qu'il reste des configurations non générés :  
        Générer(coloration)  
        si coloration == correcte  
            retourner coloration  
    fin tant_que  
Fin
```

Complexité :  $\Omega(\text{nb\_couleurs} \wedge \text{nb\_sommets})$

Nous n'avons pas eu le temps de coder cet algo, nous utilisons le backtrack.

### Version Retour sur Trace :

On génère et teste toutes les instances possibles de ce problème, ce qui n'est pas du tout optimisé :

```
Fonction GeT (graphe g) :  
Début :  
    Pour chaque Sommet i dans g :  
        i.AssignerCouleur() ;  
    fin_pour  
fin  
  
Fonction AssignerCouleur(Sommet this) :  
Début :  
    Pour chaque couleur :  
        Si aucun voisin n'a la même couleur :  
            this.couleur = couleur  
        fin_si  
    fin_pour  
fin
```

Complexité théorique :  $\Omega(\text{nb\_couleurs} \wedge \text{nb\_sommets})$ .

Cet algo a la même complexité théorique que le GeT, cependant, en pratique, ReT a une complexité temporelle plus faible, et une complexité spatiale très inférieure car on ne stocke pas tous les résultats.

## Algo génère SAT:

On ajoute 3 sommets a, b, c.

Pour chaque clause, on génère un

Fonction

On ajoute 3 sommets a, b, c.

relier (a, b)

relier(b, c)

relier(a, c)

Pour chaque variable n :

    Ajouter 2 sommets :

    x1 [n], \_x1[n]

Pour chaque clause L :

    Ajouter 5 sommets : D[L], E[L], F[L], G[L], H[L]

    relier (D[L], E[L])

    relier (D[L], F[L])

    relier (F[L], G[L])

    relier (G[L], H[L])

    relier (F[L], H[L])

    relier (F[L], H[L])

    relier (tous les x, G[L])

    relier (tous les \_x, H[L])

fin\_pour

relier (C, tous les x)

relier (C, tous les \_x)

relier (A, tous les D)

relier (A, tous les E)