**Analyse en complexité de temps et d’espace :**

FileSystem.readFile -> O(n+m)

Algo.coloringRec(graph) -> n^2log(n) + 4n^3 + 2mn^2 + 2n^3log(n) -> O(n^3\*log(n))

GetUnusedColorsInNeighbours -> { O(nlog(n))}

Brique 4 -> {O(n)}

Brique 5 -> {O(1)}

Brique 6 -> {O(n^2 +n^2+ nm + n^2log(n))} -> O(n^2)

Brique 5’ -> {O(1)}

Brique 6’ -> {O(n^2 +n^2+ nm + n^2log(n))} -> O(n^2)

CheckColoring -> O(n+m)

FileSystem.writefile -> O(n)

Affichage des couleurs -> O(n)

ReadGraphicFile -> O(n+m)

Donc la complexité en temps et en espace de notre implémentation est :

O(n+m+n^3log(n) + n + m + n +n +n +m) -> **O(n^3 \* log(n))**

* **O(n^3 \* log(n))**

**Temps d’exécution :**

Les temps donnés sont des approximations :

Pour 10 sommets :

Temps d'exécution (sans le graphique) : 0.004484699 secondes

Temps d'exécution (avec le graphique) : 0.1494323 secondes

Pour 12 sommets :

Temps d'exécution (sans le graphique) : 0.0081986 secondes

Temps d'exécution (avec le graphique) : 0.1766408 secondes

Pour 50 sommets :

Temps d'exécution (sans le graphique) : 0.008333 secondes

Temps d'exécution (avec le graphique) : 0.1851763 secondes

Pour 100 sommets :

Temps d'exécution (sans le graphique) : 0.01262 secondes

Temps d'exécution (avec le graphique) : 0.1755862 secondes

**Choix d’implémentations, difficultés, solutions :**

Nous avons décidé d’implémenter la version récursive de la 5-coloration des graphes, en effet, lors de notre travaille la difficulté majeure auxquels nous avons fait face a été l’implémentation de la brique 6. En effet, la difficulté venait du fait que nous n’avions pas entièrement compris ce que nous expliquais la brique 6.

Après avoir passé un moment à réfléchir, nous avons trouvé une solution pour résoudre ce problème. Pour implémenté la brique 6 :

Soit un sommet x, nous récupérons les voisins y de x et pour chaque y nous récupérons la composante connexe associée à y, après avoir obtenu la composante connexe nous vérifions si un y’ n’appartient pas à la composante connexe de y, si c’est le cas nous donnons la couleur de y’ à x et y’ prends la même couleur que y.

Ainsi nous avons résolu le problème rencontré avec la brique 6.

Une autre difficulté à été de gérer les sommets qui possédaient plus de 5 voisins, pour cela, nous avons décidé de donner à ces sommets une couleur qui n’a pas été utilisées dans leurs voisinages sinon nous essayons d’appliquer la méthode que la brique 6.