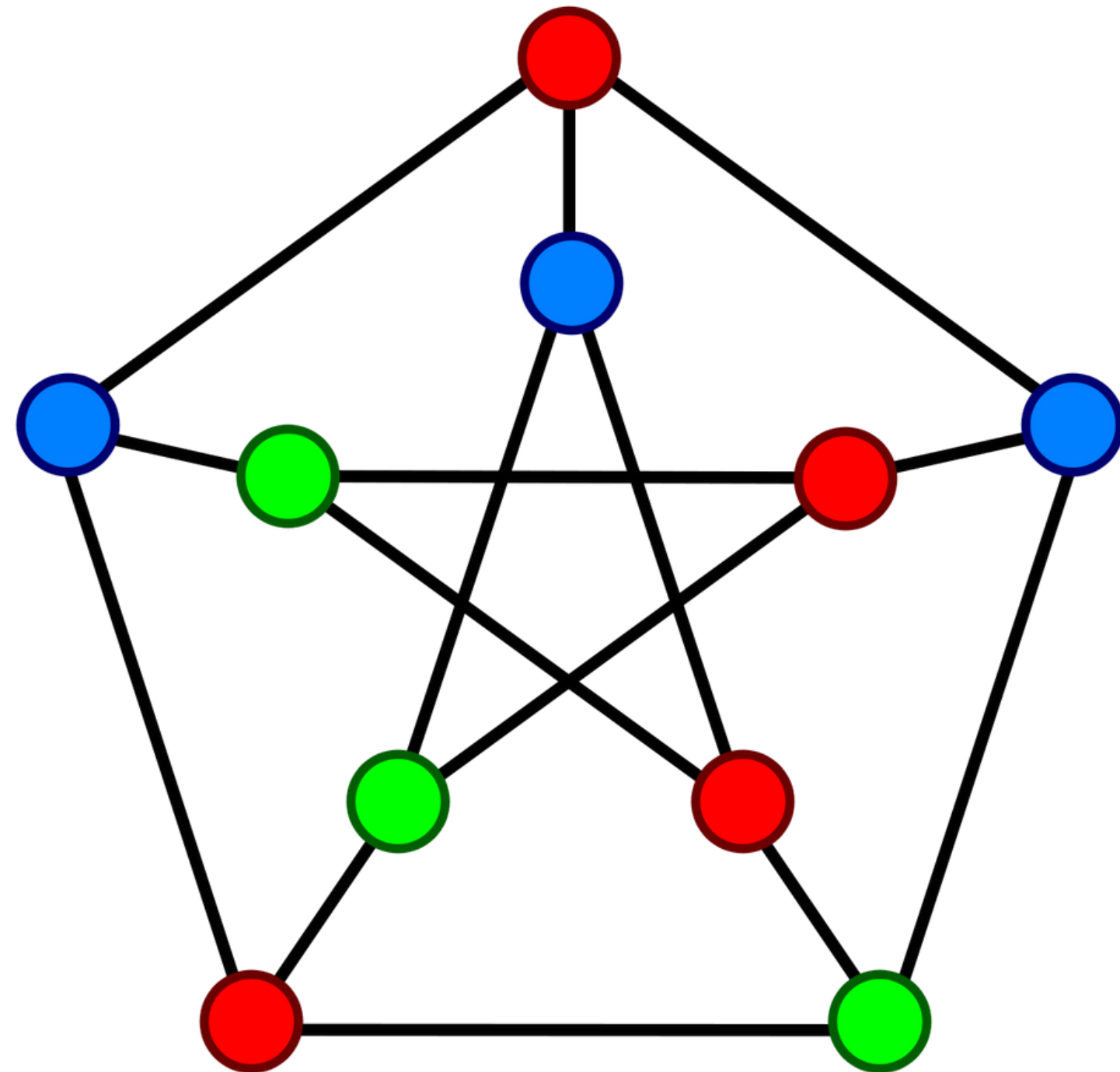


Application d'un algorithme approché pour résoudre le problème de coloration de graphe



00

L'algorithme choisi

Pour résoudre notre problème de coloration de graphe, j'ai choisi l'algorithme de Glouton.

Appelé aussi algorithme gourmand, il suit le principe de faire, étape par étape, un choix optimum local, dans l'espoir d'obtenir un résultat optimum global.

01

Le Pseudo-code de l'algorithme

Pseudo-code

Pour la coloration d'un graphe $G = (X, A)$ avec $|X| = n$.

liste ordonnée de sommets $X = [x_1, x_2, \dots, x_n]$.

liste ordonnée de couleurs C .

Pour i de 1 à n faire

Sommet = x_i

Couleur = la couleur minimale de C non utilisée par les voisins de **Sommet**

 Affecter à **Sommet** la couleur **Couleur**.

Fin pour.

Afficher le nombre de couleurs utilisés.

02

L'utilité des différents paramètres

Les paramètres

Pour colorer un graphe $G = (x, A)$, on a comme paramètres:

Un nombre chromatique $x(G)$:

Le nombre chromatique d'un graphe G (orientée ou non orientée) est le plus petit nombre de couleurs suffisant pour colorer le graphe G .
Le nombre chromatique d'un graphe G est en général noté $x(G)$.

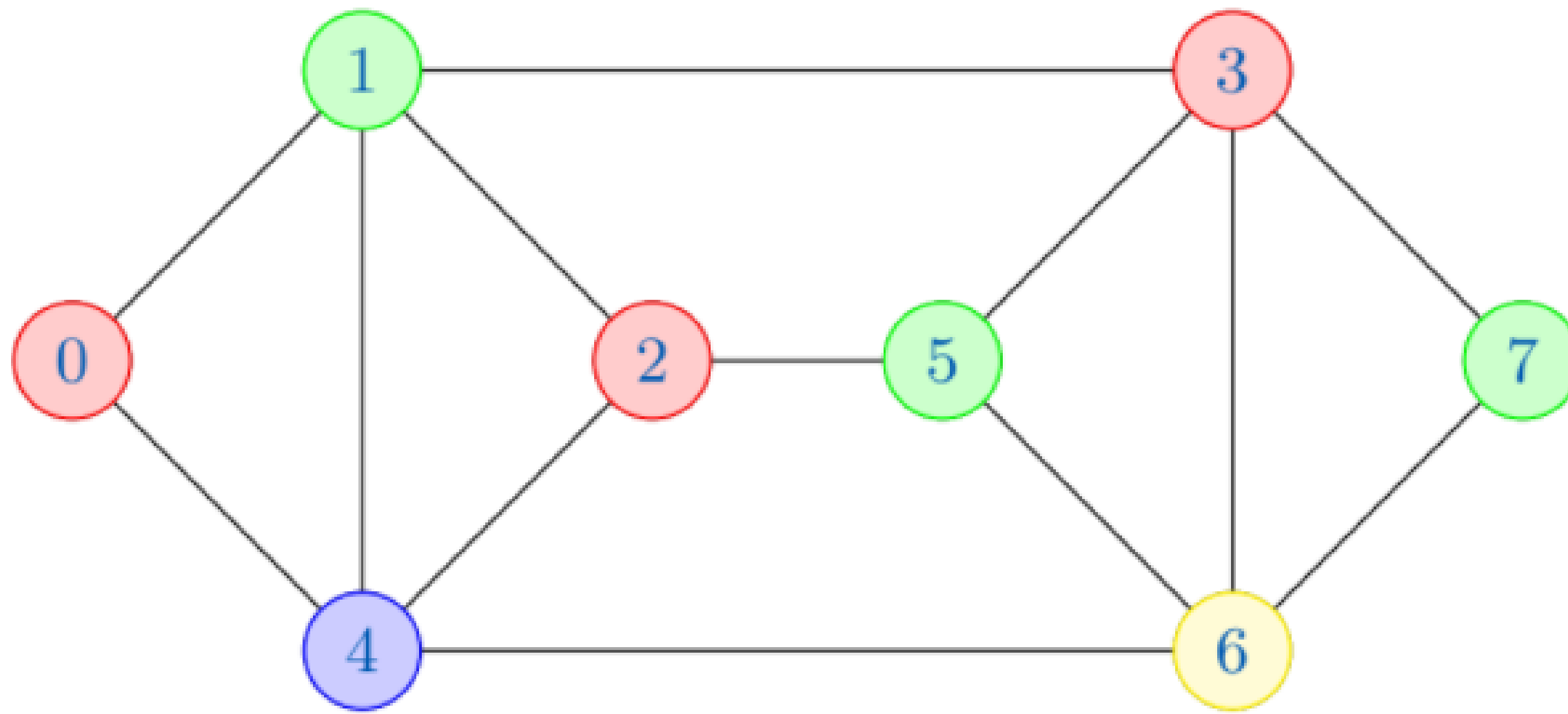
liste ordonnée de sommets $X = [x_1, x_2, \dots, x_n]$.

liste ordonnée de couleurs C .

03

**L'application de
l'algorithme pour résoudre
notre problème**

Cet algorithme propose une coloration qui sera valide, mais pas forcément optimale, et ce n'est donc qu'une heuristique.



Numéro	Couleur
c_1	rouge
c_2	vert
c_3	bleu
c_4	jaune