

Nom :

Prénom :

Groupe :

Test TP CSP (Sujet N°1)

Exercice

On souhaite faire la coloration d'un graphe de la figure 1, la coloration consiste à assigner à chaque sommet du graphe une couleur de telle sorte qu'aucune paire de sommets adjacents n'aient la même couleur. On dispose pour cela des 3 couleurs suivantes : bleu, rouge et vert.

- 1) Modéliser le problème de coloration du graphe la figure 1 sous la forme d'un CSP. (2pts)

- 2) Donner une structure de données permettant de modéliser le graphe de la figure 1, puis donner la structure correspondante (2pt)

- 3) Écrire une fonction permettant de vérifier la consistance d'une assignation après l'ajout d'une nouvelle assignation. (2pts)
- 4) Implémenter l'algorithme de **back-tracking**. (2.5pts)
- 5) Créer une fonction permettant de vérifier la validité d'une solution (2pts)
- 6) Tester le fonctionnement de votre algorithme (1pt)
- 7) Réaliser un **test unitaire** permettant de valider le fonctionnement de votre algorithme (3pts)
- 8) Compresser votre travail dans une archive sous le nom **nom_prenom_groupe.zip** (0.5pt)
- 9) Perfection de code (1pt)

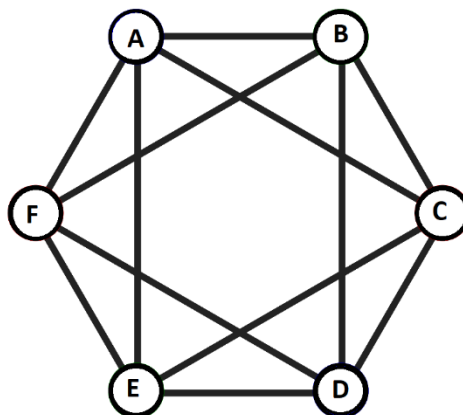


Figure 1 : Le graphe à colorer

Nom :

Prénom :

Groupe :

Test TP CSP (Sujet N°2)

Exercice

On souhaite faire la coloration d'un graphe de la figure 2, la coloration consiste à assigner à chaque sommet du graphe une couleur de telle sorte qu'aucune paire de sommets adjacents n'aient la même couleur. On dispose pour cela des 3 couleurs suivantes : bleu, rouge et vert.

- 1) Modéliser le problème de coloration du graphe la figure 2 sous la forme d'un CSP. (2pts)

- 2) Donner une structure de données permettant de modéliser le graphe de la figure 2, puis donner la structure correspondante (2pt)

- 3) Écrire une fonction permettant de vérifier la consistance d'une assignation après l'ajout d'une nouvelle assignation. (2pts)
- 4) Implémenter l'algorithme de **back-tracking**. (2.5pts)
- 5) Créer une fonction permettant de vérifier la validité d'une solution (2pts)
- 6) Tester le fonctionnement de votre algorithme (1pt)
- 7) Réaliser un **test unitaire** permettant de valider le fonctionnement de votre algorithme (3pts)
- 8) Compresser votre travail dans une archive sous le nom **nom_prenom_groupe.zip** (0.5pt)
- 9) Perfection de code (1pt)

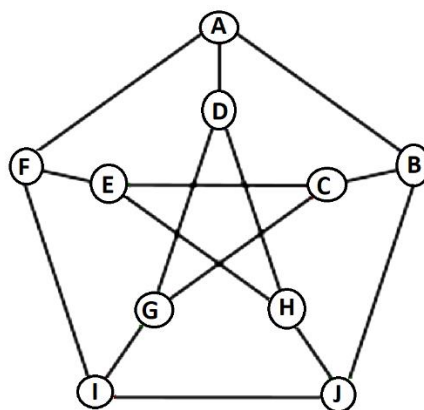


Figure 2 : Le graphe à colorer

Nom :

Prénom :

Groupe :

Test TP CSP (Sujet N°3)

Exercice

On souhaite faire la coloration d'un graphe de la figure 3, la coloration consiste à assigner à chaque sommet du graphe une couleur de telle sorte qu'aucune paire de sommets adjacents n'aient la même couleur. On dispose pour cela des 3 couleurs suivantes : bleu, rouge et vert.

- 1) Modéliser le problème de coloration du graphe la figure 3 sous la forme d'un CSP. (2pts)

- 2) Donner une structure de données permettant de modéliser le graphe de la figure 3, puis donner la structure correspondante (2pt)

- 3) Écrire une fonction permettant de vérifier la consistance d'une assignation après l'ajout d'une nouvelle assignation. (2pts)
- 4) Implémenter l'algorithme de **back-tracking**. (2.5pts)
- 5) Créer une fonction permettant de vérifier la validité d'une solution (2pts)
- 6) Tester le fonctionnement de votre algorithme (1pt)
- 7) Réaliser un **test unitaire** permettant de valider le fonctionnement de votre algorithme (3pts)
- 8) Compresser votre travail dans un archive sous le nom **nom_prenom_groupe.zip** (0.5pt)
- 9) Perfection de code (1pt)

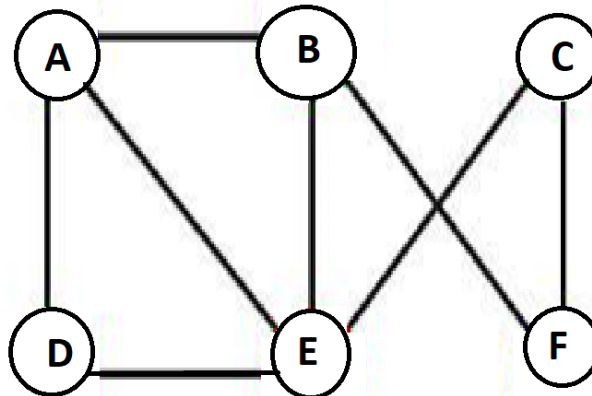


Figure 3 : Le graphe à colorer

Nom :

Prénom :

Groupe :

Test TP CSP (Sujet N°4)

Exercice

On souhaite faire la coloration d'un graphe de la figure 4, la coloration consiste à assigner à chaque sommet du graphe une couleur de telle sorte qu'aucune paire de sommets adjacents n'aient la même couleur. On dispose pour cela des 3 couleurs suivantes : bleu, rouge et vert.

- 1) Modéliser le problème de coloration du graphe la figure 4 sous la forme d'un CSP. (2pts)

- 2) Donner une structure de données permettant de modéliser le graphe de la figure 4, puis donner la structure correspondante (2pt)

- 3) Écrire une fonction permettant de vérifier la consistance d'une assignation après l'ajout d'une nouvelle assignation. (2pts)
- 4) Implémenter l'algorithme de **back-tracking**. (2.5pts)
- 5) Créer une fonction permettant de vérifier la validité d'une solution (2pts)
- 6) Tester le fonctionnement de votre algorithme (1pt)
- 7) Réaliser un **test unitaire** permettant de valider le fonctionnement de votre algorithme (3pts)
- 8) Compresser votre travail dans un archive sous le nom **nom_prenom_groupe.zip** (0.5pt)
- 9) Perfection de code (1pt)

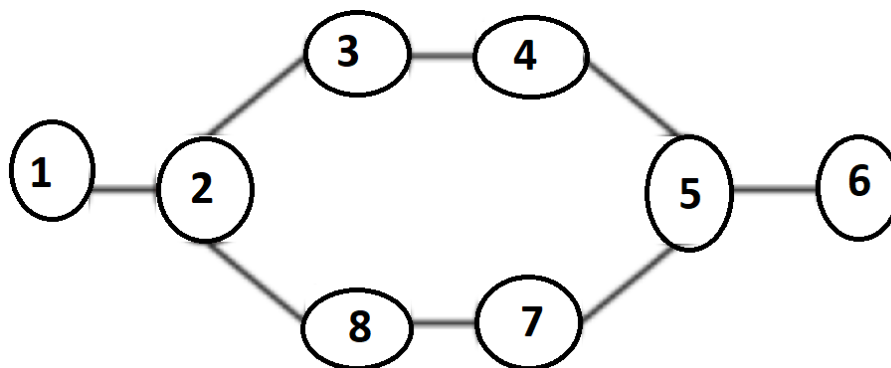


Figure 4 : Le graphe à colorer

Nom :

Prénom :

Groupe :

Test TP CSP (Sujet N°5)

Exercice

On souhaite faire la coloration d'un graphe de la figure 5, la coloration consiste à assigner à chaque sommet du graphe une couleur de telle sorte qu'aucune paire de sommets adjacents n'aient la même couleur. On dispose pour cela des 3 couleurs suivantes : bleu, rouge et vert.

- 1) Modéliser le problème de coloration du graphe la figure 5 sous la forme d'un CSP. (2pts)

- 2) Donner une structure de données permettant de modéliser le graphe de la figure 5, puis donner la structure correspondante (2pt)

- 3) Écrire une fonction permettant de vérifier la consistance d'une assignation après l'ajout d'une nouvelle assignation. (2pts)
- 4) Implémenter l'algorithme de **back-tracking**. (2.5pts)
- 5) Créer une fonction permettant de vérifier la validité d'une solution (2pts)
- 6) Tester le fonctionnement de votre algorithme (1pt)
- 7) Réaliser un **test unitaire** permettant de valider le fonctionnement de votre algorithme (3pts)
- 8) Compresser votre travail dans un archive sous le nom **nom_prenom_groupe.zip** (0.5pt)
- 9) Perfection de code (1pt)

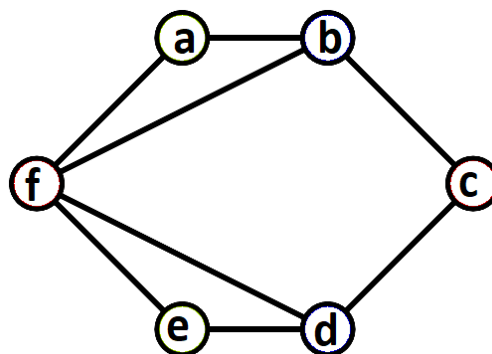


Figure 5 : Le graphe à colorer

Nom :

Prénom :

Groupe :

Test TP CSP (Sujet N°6)

Exercice

On souhaite faire la coloration d'un graphe de la figure 6, la coloration consiste à assigner à chaque sommet du graphe une couleur de telle sorte qu'aucune paire de sommets adjacents n'aient la même couleur. On dispose pour cela des 3 couleurs suivantes : bleu, rouge et vert.

- 1) Modéliser le problème de coloration du graphe la figure 6 sous la forme d'un CSP. (2pts)

- 2) Donner une structure de données permettant de modéliser le graphe de la figure 6, puis donner la structure correspondante (2pt)

- 3) Écrire une fonction permettant de vérifier la consistance d'une assignation après l'ajout d'une nouvelle assignation. (2pts)
- 4) Implémenter l'algorithme de **back-tracking**. (2.5pts)
- 5) Créer une fonction permettant de vérifier la validité d'une solution (2pts)
- 6) Tester le fonctionnement de votre algorithme (1pt)
- 7) Réaliser un **test unitaire** permettant de valider le fonctionnement de votre algorithme (3pts)
- 8) Compresser votre travail dans un archive sous le nom **nom_prenom_groupe.zip** (0.5pt)
- 9) Perfection de code (1pt)

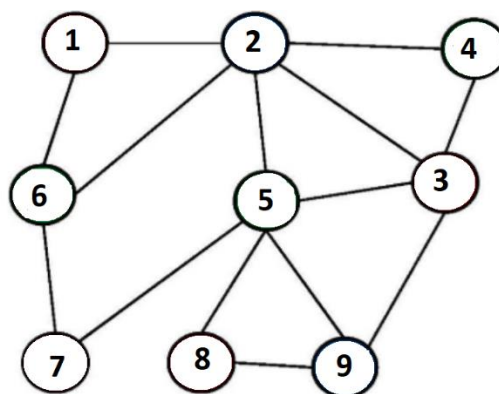


Figure 6 : Le graphe à colorer