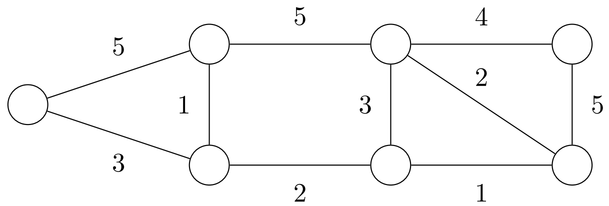
Exercice 1. Kruskal et Prim

Appliquer l'algorithme de Kruskal puis l'algorithme de Prim sur le graphe suivant :

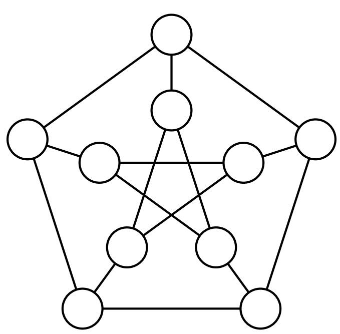


Exercice 2. Coloration de graphe

Soit $G = (V, E)$ un graphe. Une k-**coloration** de G est une fonction $c : V \longmapsto \{0, ..., k-1\}$ vérifiant :

$$\forall \{u, v\} \in E,\ c(u) \neq c(v)$$

1. Trouver une 3-coloration du graphe de Petersen :



2. Montrer que ce graphe n'est pas 2-coloriable.

On appelle le **nombre chromatique** $\chi(G)$ le plus petit k tel que G soit k-coloriable. Par exemple, le nombre chromatique du graphe de Petersen est 3.

On considère un algorithme glouton de coloriage :

```
C ⟵ ∅
Pour chaque sommet v (dans un ordre quelconque) :
  Si une couleur de C n'est utilisé par aucun voisin de v :
    Donner à v cette couleur
  Sinon :
    Ajouter une nouvelle couleur à C et l'utiliser pour v
```

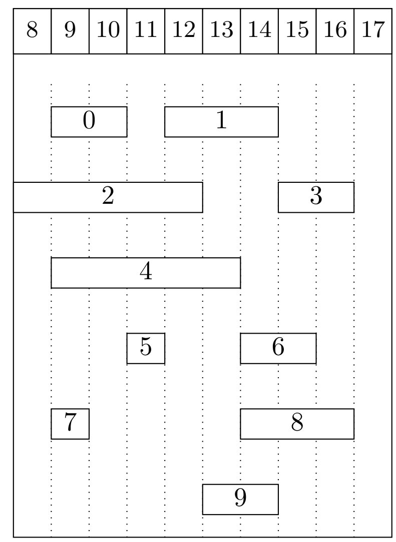
3. Montrer sur un exemple simple que le coloriage obtenu n'est pas forcément optimal.

On souhaite créer un emploi du temps pour une journée : chaque cours possède une heure de début et de fin et doit être assigné à une salle. Il ne peut pas y avoir 2 cours en même temps dans la même salle. L'objectif est de minimiser le nombre de salles à utiliser.

4. Modéliser ce problème sous forme de coloriage de graphe.

Dans l'algorithme glouton précédent, en considère maintenant les sommets par ordre de début de cours croissant (on regarde d'abord le cours qui se termine le plus tôt).

5. Appliquer ce nouvel algorithme glouton sur :



6. Soit k le nombre maximum de cours se déroulant pendant la même heure. Montrer que le nombre chromatique du graphe de la question 4 est au moins k.
7. Montrer que l'algorithme glouton donne un coloriage a k couleurs et est donc optimal.

On dit qu'un graphe est **biparti** s'il ne possède pas de cycle de longueur impaire.

8. Montrer qu'un graphe biparti est 2-coloriable en donnant un algorithme pour trouver un 2-coloriage.

Exercice 3. Approximation du TSP

Soit $G = (V, E)$ un graphe complet (toutes les arêtes possibles existent) pondéré par w vérifiant :

$$\forall, x, y, z \in V,\ w(x, z) \leq w(x, y) + w(y, z)$$

Le problème du voyageur de commerce (TSP) consiste à trouver un cycle de poids minimum w^* visitant tous les sommets exactement une fois.
Soit T un arbre couvrant de poids minimum de G.

1. Montrer que $w(T) \leq w^*$.

On duplique ensuite chaque arête de T pour obtenir T', vérifiant $w(T') = 2w(T) \leq 2w^*$.

2. Comment obtenir un cycle C visitant tous les sommets exactement une fois à partir de $w(T')$?
3. Montrer que $w(C) \leq 2w^*$ (c'est une 2-approximation du TSP).