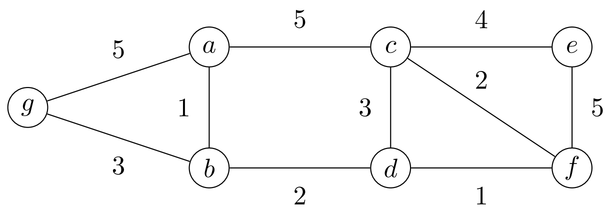
Exercice 1. Kruskal et Prim

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



Solution : On trouve dans les 2 cas un arbre couvrant minimum de poids 13

- **Kruskal** : on choisit dans l'ordre les arêtes (a, b), (d, f), (d, b), (c, f), (g, b), (c, e)
- **Prim**, en partant par exemple de a : (a, b), (b, d), (d, f), (f, c), (b, g), (c, e)

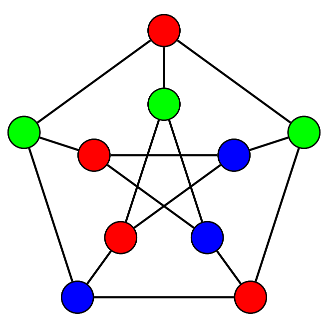
Exercice 2. Coloration de graphe

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 :

   Solution :



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

   Solution : Le graphe de Petersen contient plusieurs cycles de longueur 5. On remarque facilement qu'il n'est pas possible de colorier un tel cycle avec 2 couleurs (ni d'ailleurs n'importe quel cycle de longueur impaire).

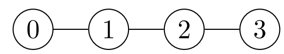
On appelle le **nombre chromatique** $\chi(G)$ le plus petit k tel que G soit k-coloriable. Par exemple, le nombre chromatique du graphe de Petersen est 3.

On considère un algorithme glouton de coloriage :

$C \longleftarrow \emptyset$
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.

   Solution : Si on colorie un chemin à 4 sommets en commençant par les deux extrêmités, l'algorithme glouton va leur donner la même couleur. Il va ensuite être obligé de donner 2 nouvelles couleurs aux deux autres sommets, ce qui donne 4 couleurs qui n'est pas l'optimum (on peut le colorier avec 2 couleurs).



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.

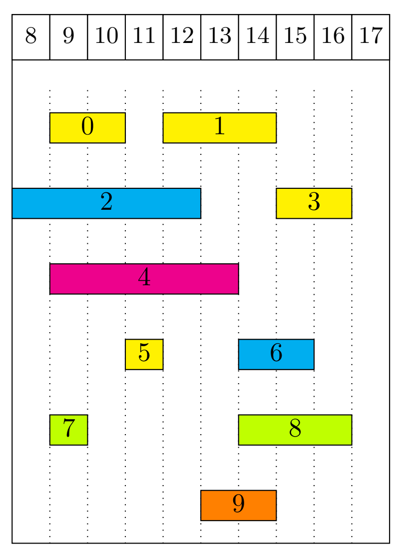
4. Modéliser ce problème sous forme de coloriage de graphe.

   Solution : On définit $G = (V, E)$ où
   - V est l'ensemble des cours (chaque sommet est un intervalle de temps)
   - E représente les chevauchements de cours (deux sommets sont reliés si les intervalles de temps s'intersectent)

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 le diagramme de Gantt suivant :

   Solution : On trouve qu'un a besoin de 4 salles différentes.



6. Soit k le nombre maximum de cours se déroulant pendant la même heure. Montrer que le nombre chromatique du graphe de la question 5 est au moins k.

> **Solution** : Il faut colorier chacun de ces cours avec une couleur différente.

7. Montrer que l'algorithme glouton donne un coloriage à k couleurs et est donc optimal.

> **Solution** : Supposons par l'absurde que l'algorithme glouton utilise au moins $k+1$ couleurs et considérons le premier intervalle I pour lequel l'algorithme utilise $k+1$ couleurs.
> Soit i l'heure de début de I. Comme l'algorithme utilise $k+1$ couleurs pour I, il existe k intervalles déjà coloriés qui intersectent I. De plus ces intervalles commencent avec i car ils ont été considérés avant dans l'algorithme. Donc il y a $k+1$ intervalles qui contiennent i.
> Ceci contredit la définition de k.

8. Est-ce que l'algorithme glouton serait optimal si on triait par ordre de **début croissant**?

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

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

> **Solution** : On fait un parcours en largeur depuis un sommet de départ s quelconque, en donnant la couleur 0 à s, 1 à ses voisins, puis en alternant les couleurs.
> Par propriété de parcours en largeur, on colorie en 0 tous les sommets à distance paire de s et en 1 tous les sommets à distance impaire. Ainsi, deux sommets u et v coloriés de la même couleur ne peuvent pas être adjacents : sinon, le chemin composé du chemin de s à u, puis de $\{u, v\}$, puis du chemin de v à s serait de longueur impaire.

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 $w(T) \leq w^*$.

> **Solution** : En enlevant une arête au cycle de poids w^*, on obtient un arbre couvrant dont le poids est supérieur à $w(T)$.

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 T'?

> **Solution** : On considère un cycle qui parcourt les sommets dans un ordre préfixe de parcours en profondeur. Cela revient à utiliser des "shortcuts" vers le prochain sommet non visité lorsqu'on est sur le point de parcourir une arête déjà visitée.

3. Montrer que $w(C) \leq 2w^*$ (c'est une 2-approximation du TSP).

4. **Solution** : Le cycle précédent est de poids inférieur à $w(T')$ d'après l'inégalité triangulaire supposée dans l'énoncé. D'où le résultat.