Graphes et applications aux réseaux

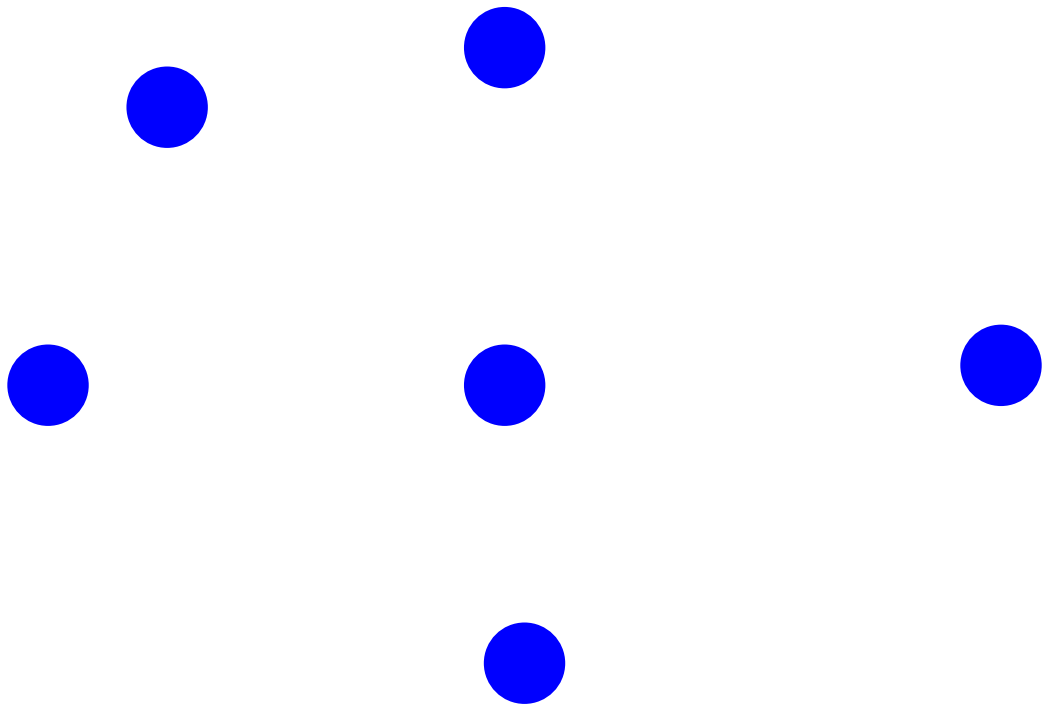
Frédéric Havet

MASCOTTE, commun I3S(CNRS/UNSA)-INRIA Sophia Antipolis

Science culture au Lycée – Lycée du Rempart – 24 novembre 2011

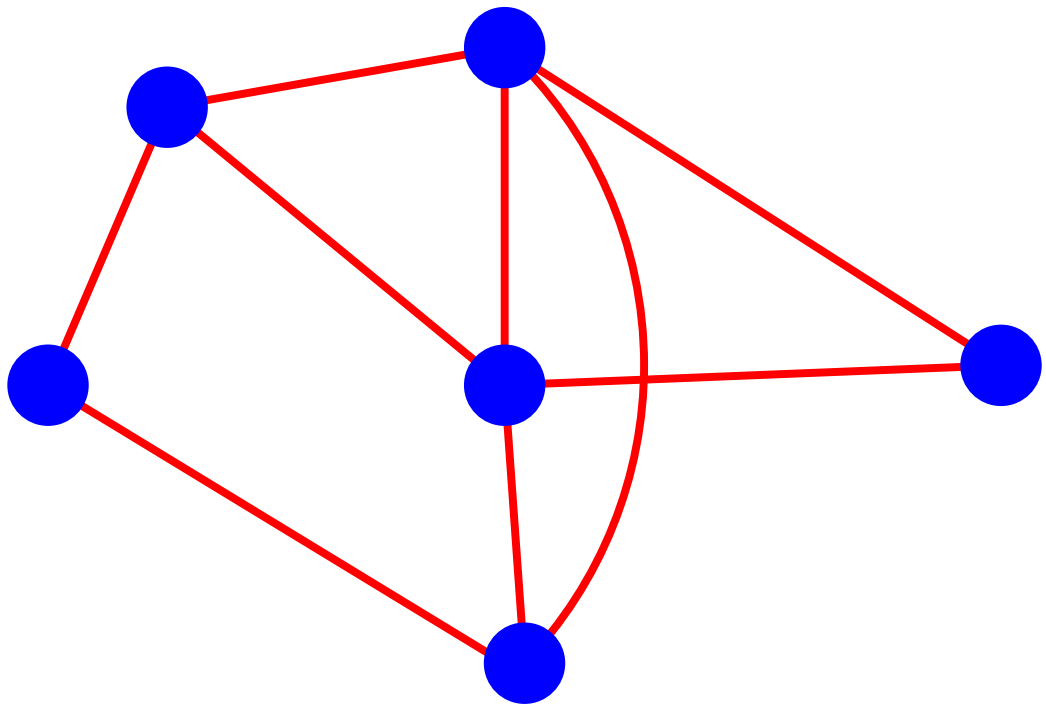
Graphe

Graphe: ensemble de sommets



Graphe

Graphe: ensemble de sommets reliés par des arêtes.

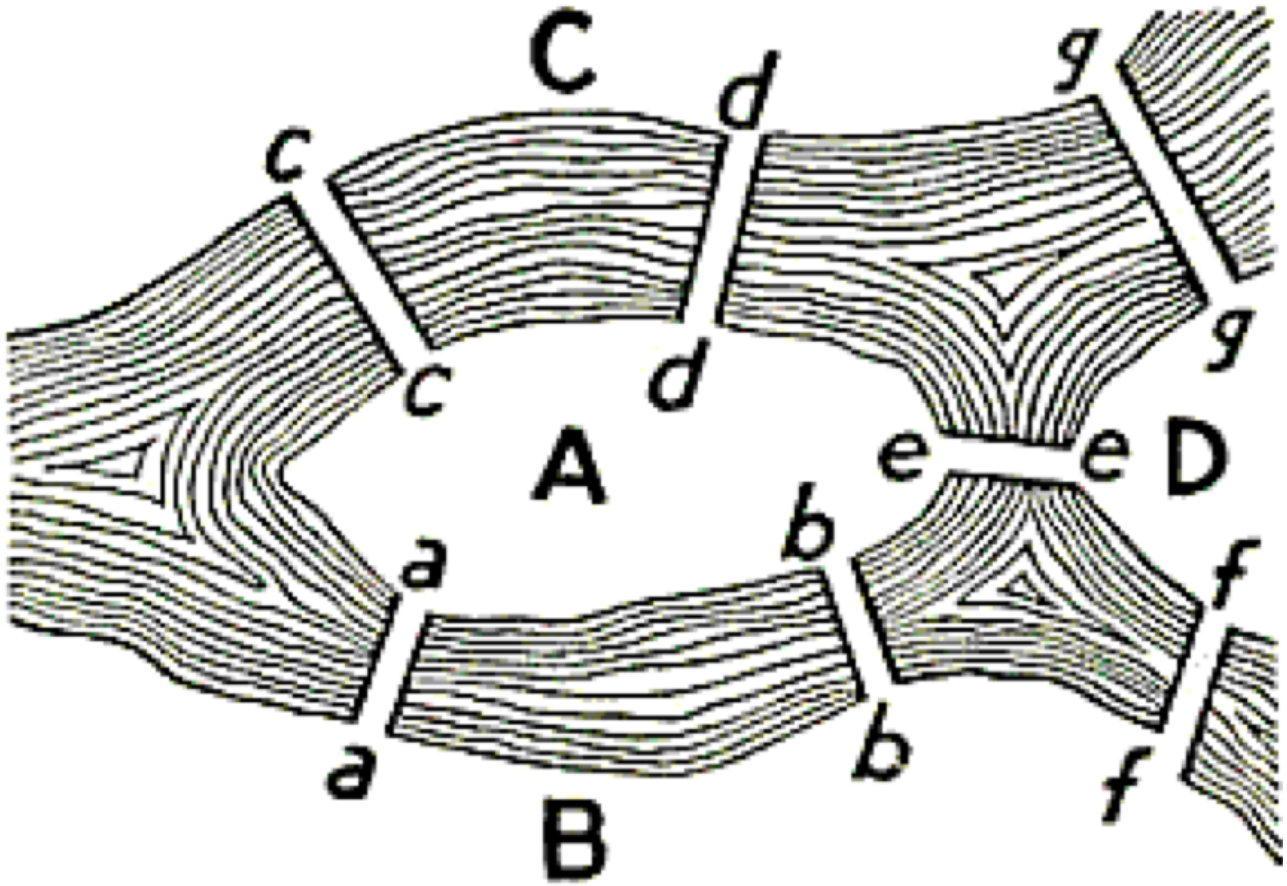


Graphe: modèle naturel pour les réseaux

- **Réseaux routiers:** Sommets $\equiv$ villes. Arêtes $\equiv$ routes.
- **Réseaux d'ordinateurs:** Sommets $\equiv$ ordinateurs/routeurs. Arêtes $\equiv$ liens physiques ou WIFI.
- **Réseaux sociaux:** Sommets $\equiv$ personnes. Arêtes entre deux amis.
- **...**

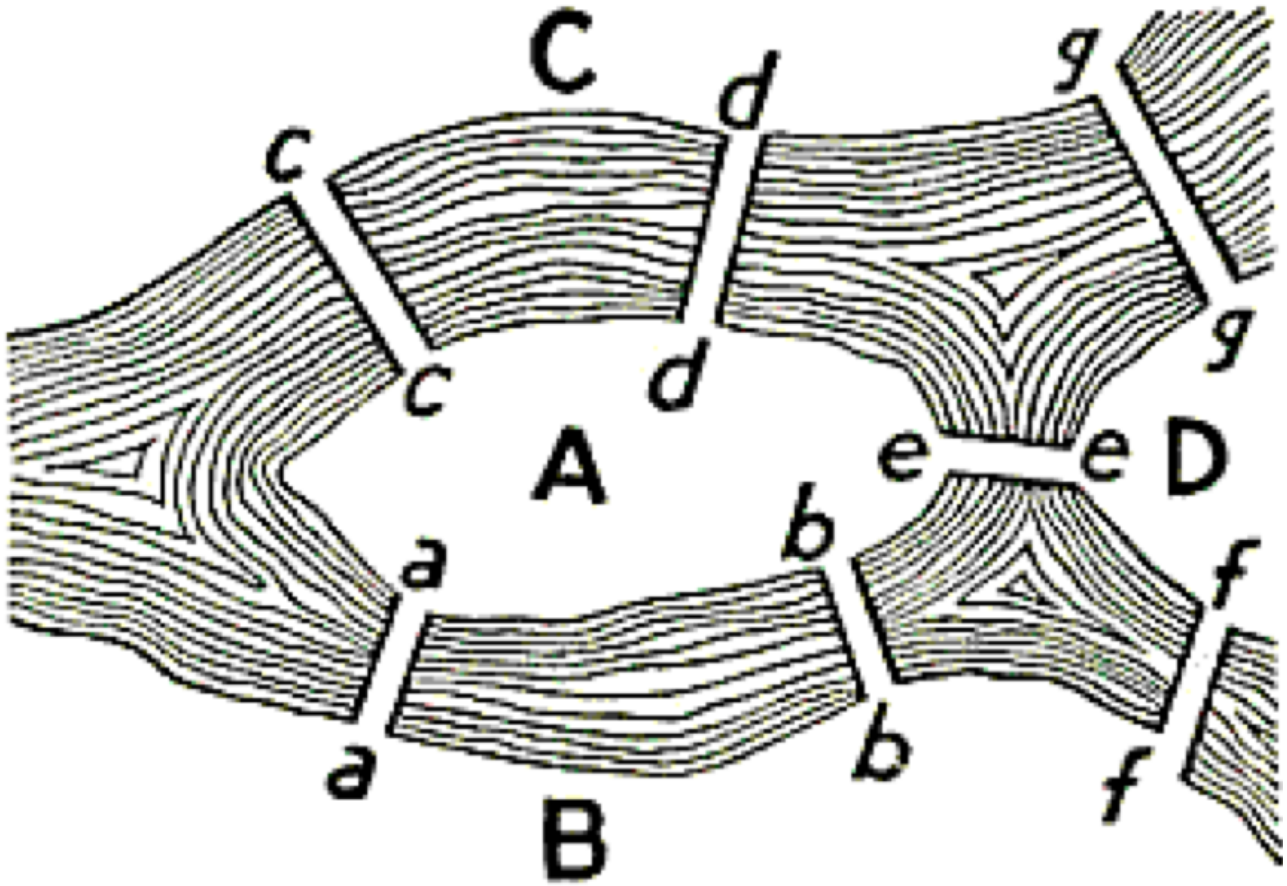
Les Ponts de Königsberg

La rivière Pregel partage la ville de Königsberg en 4 quartiers. Les habitants se demandent s'il est possible de faire un tour par chaque pont une et une seule fois. A votre avis?



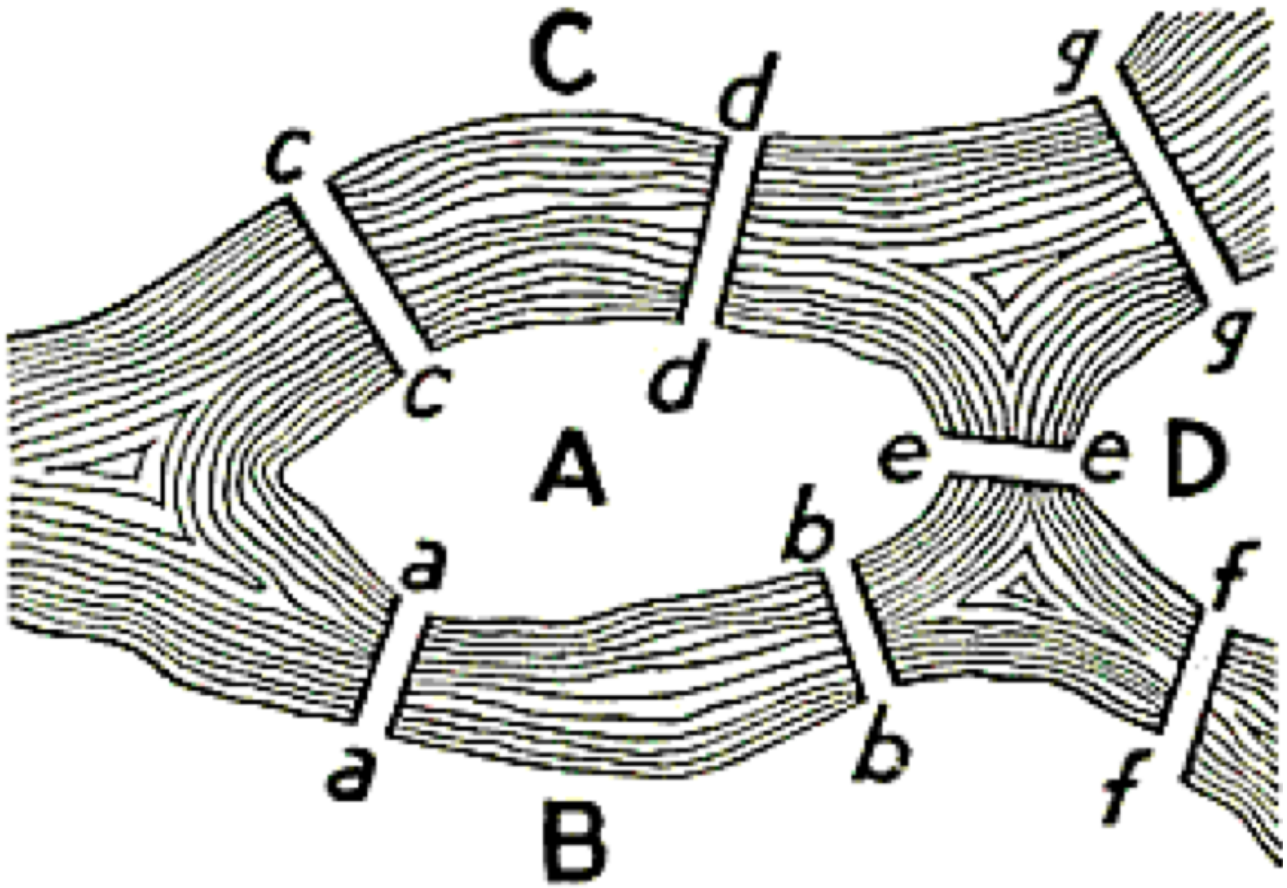
Les Ponts de Königsberg

La réponse est **NON**!
Si un tel tour existait, chaque quartier devrait être relié à un nombre **pair** de ponts. Un pont pour arriver et un pour repartir à chaque fois qu'on passe par un quartier.



Les Ponts de Königsberg

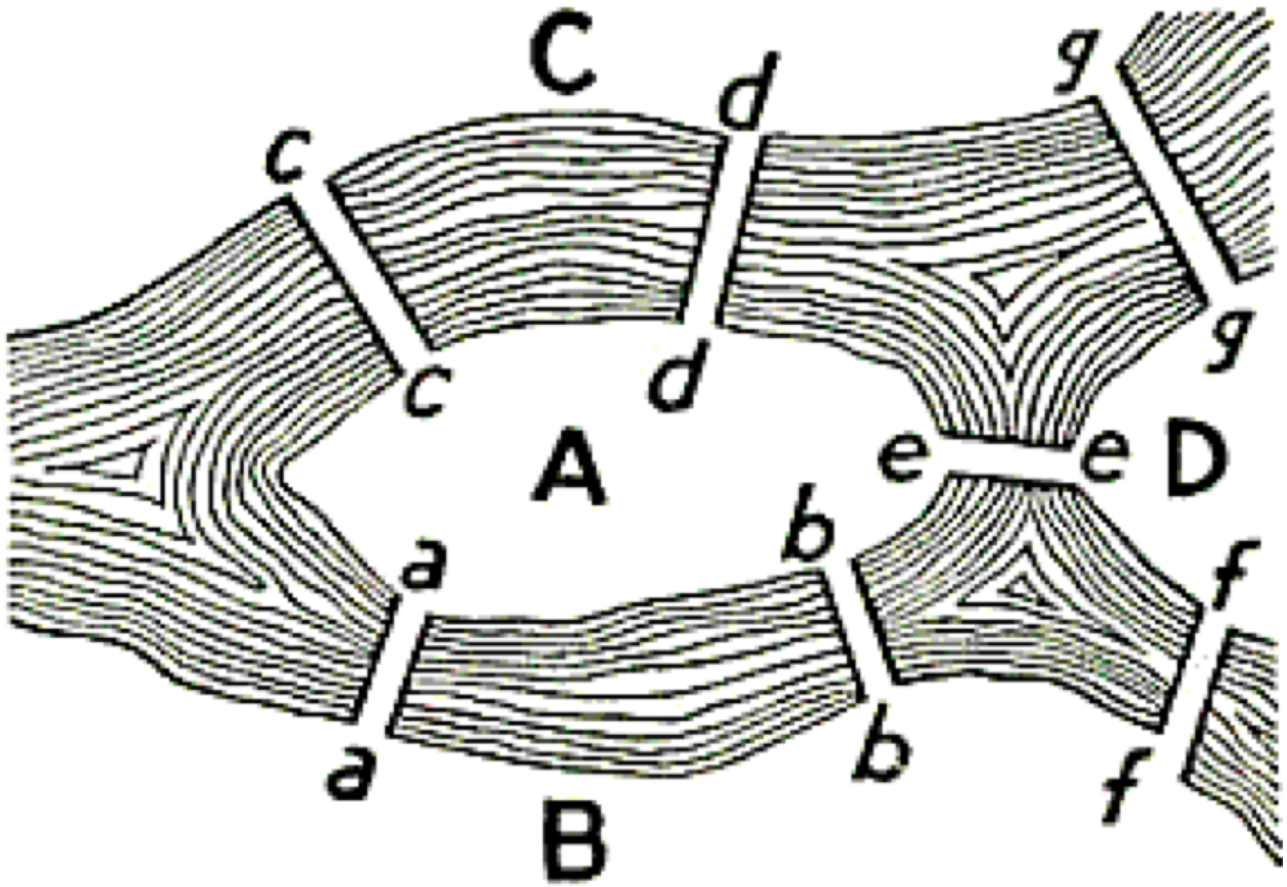
Et si on veut faire une balade (le départ et l'arrivée ne sont pas le même quartier) par chaque pont une et une seule fois?



Les Ponts de Königsberg

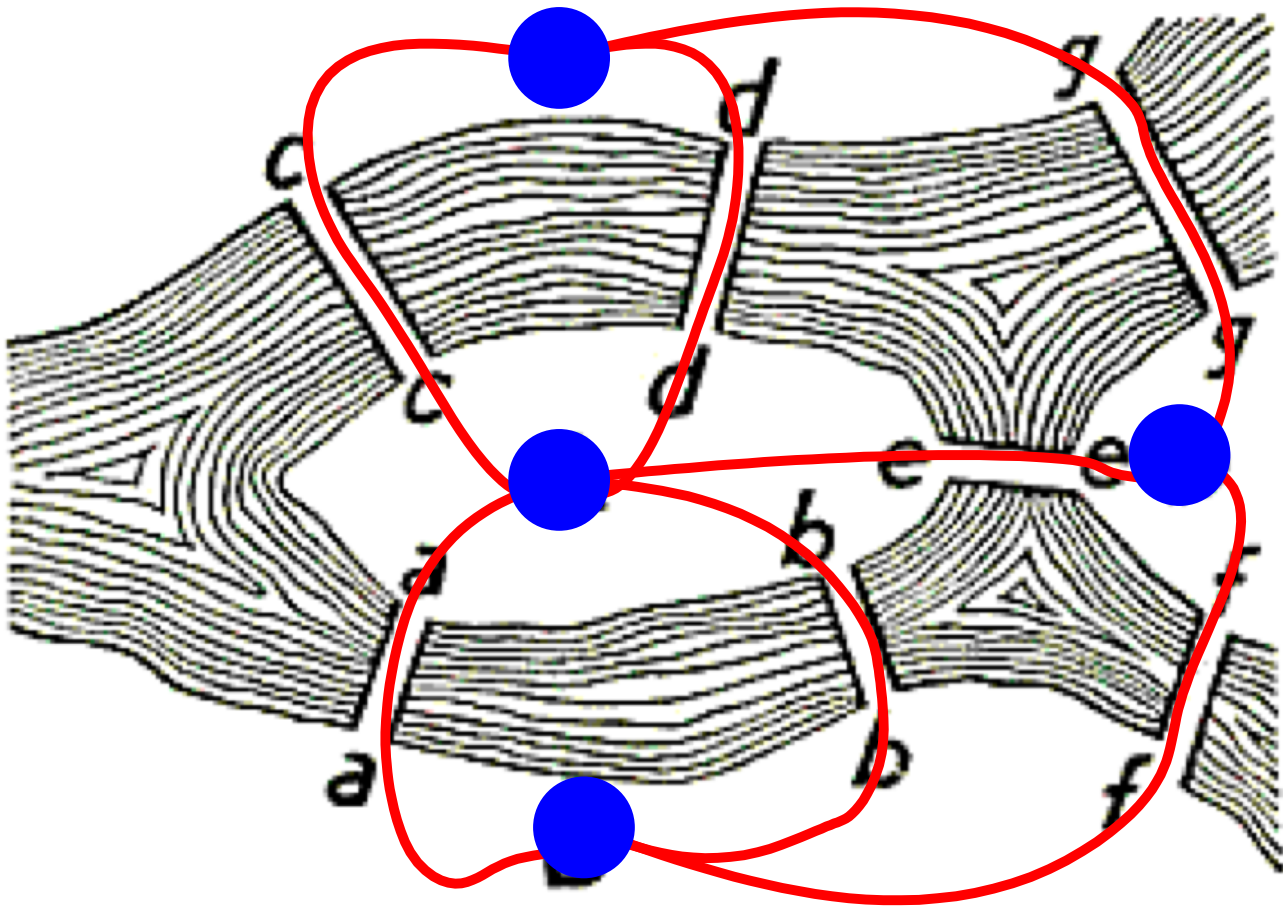
La réponse est encore **NON**!

Si une telle balade existait, tous les quartiers sauf deux (le départ et l'arrivée) devraient être reliés à un nombre pair de ponts: un pont pour arriver et un pour repartir à chaque fois qu'on passe par un quartier.



Modélisation par les graphes

Quartiers $\equiv$ sommets.
Ponts $\equiv$ arêtes.



Parcours et tour eulérien

Parcours: suite $v_0, e_1, v_1, e_2, v_2 \ldots, e_n, v_n$ telle que:

$$e_i = v_{i-1}v_i \text{ pour tout } 1 \leq i \leq n;$$

Si $v_0 = v_n$, le parcours est un tour.

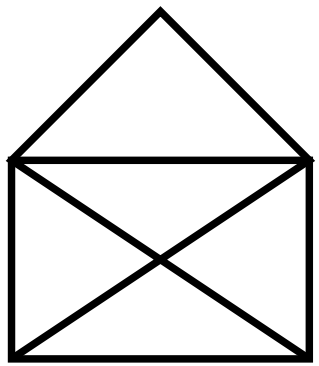
Degré d'un sommet v, $d(v)$, nb d'arêtes incidentes à v.

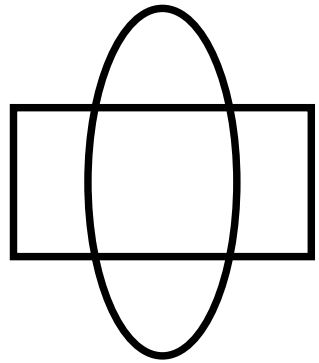
Parcours eulérien: qui passe par toutes les arêtes une et une seule fois.

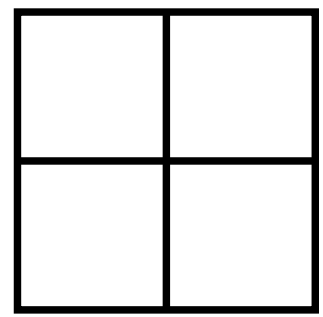
Euler: Un graphe a un parcours eulérien si et seulement si le nombre de sommets de degré impair est 0 ou 2.

Dessiner sans lever la main

Pouvez-vous dessiner chacune des figures ci-dessous sans lever le crayon et en ne passant qu'une seule fois sur chaque trait?

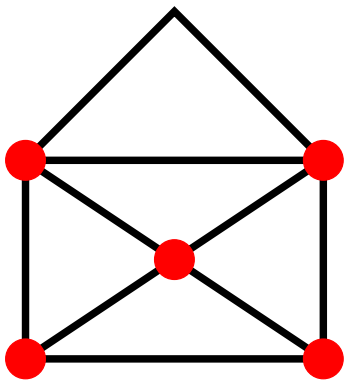


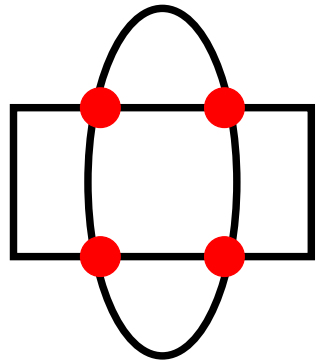


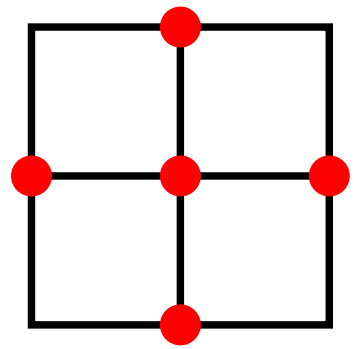


Dessiner sans lever la main

Pouvez-vous dessiner chacune des figures ci-dessous sans lever le crayon et en ne passant qu'une seule fois sur chaque trait?

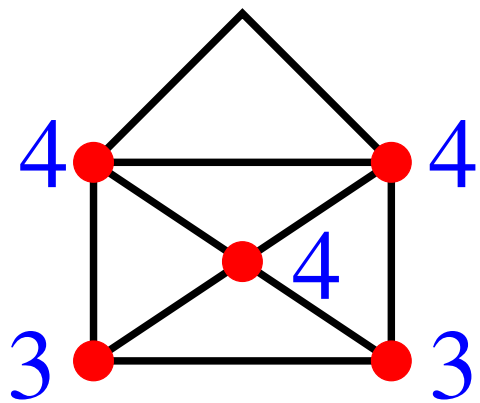


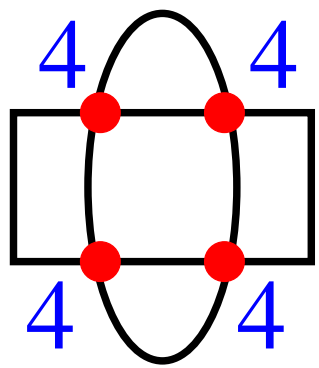


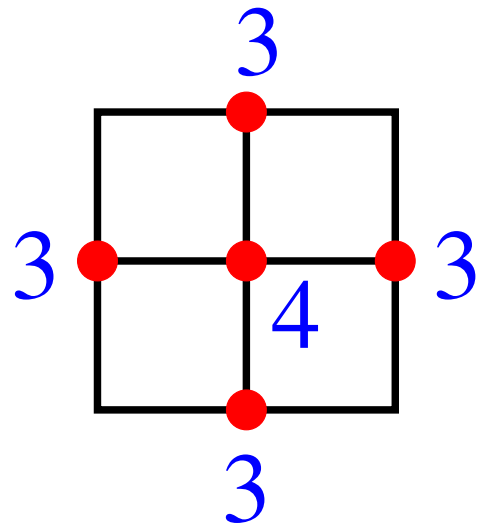


Dessiner sans lever la main

Pouvez-vous dessiner chacune des figures ci-dessous sans lever le crayon et en ne passant qu'une seule fois sur chaque trait?







Trouver un parcours eulérien

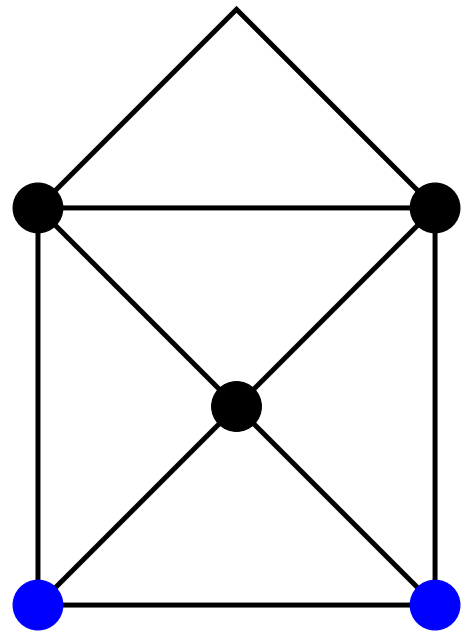
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

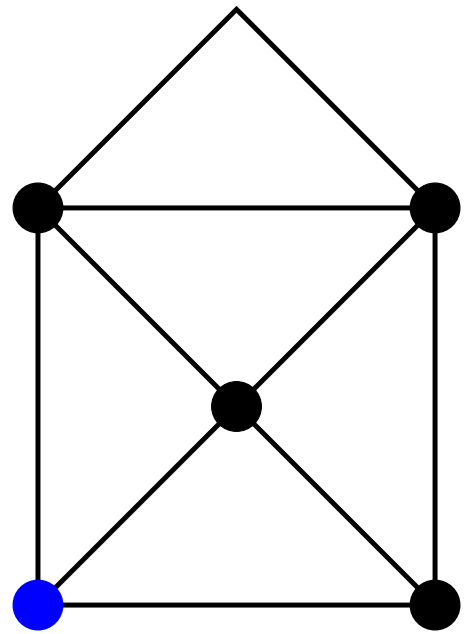
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

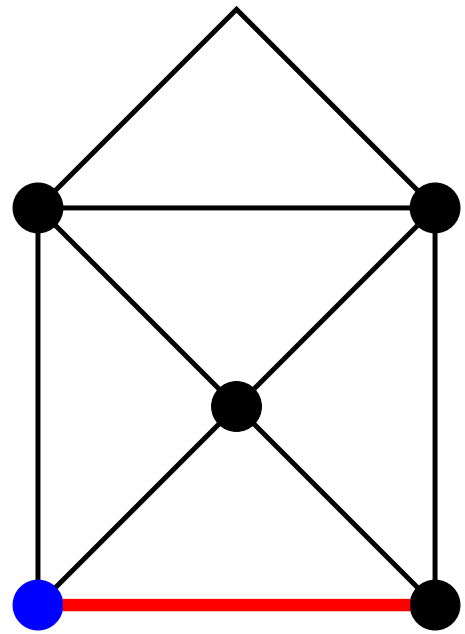
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

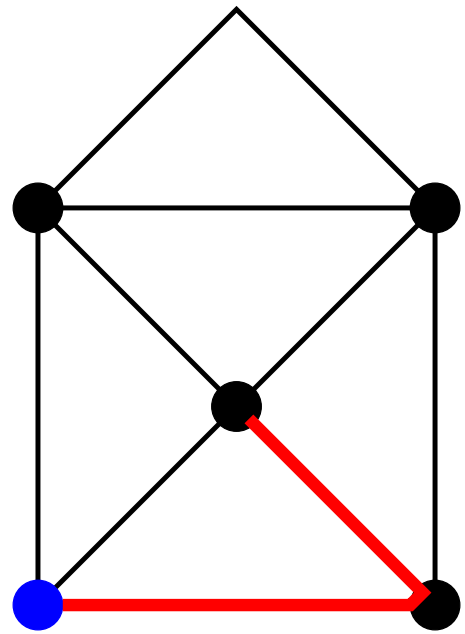
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

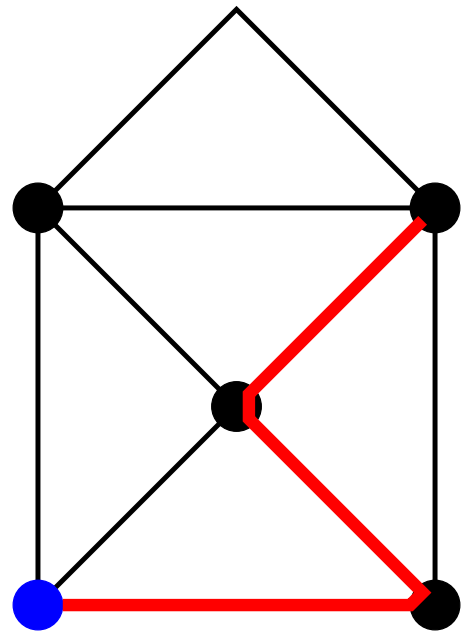
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

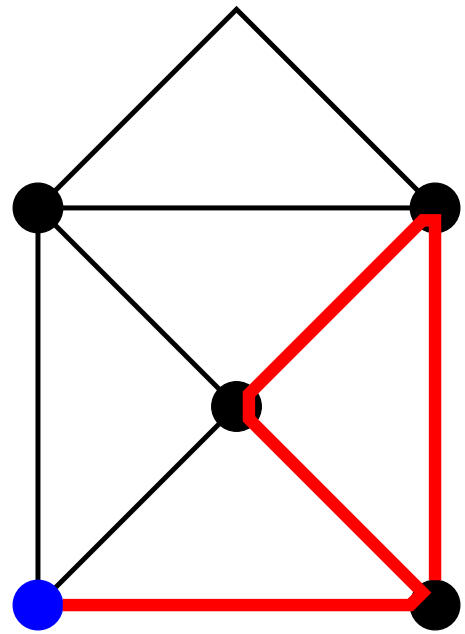
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

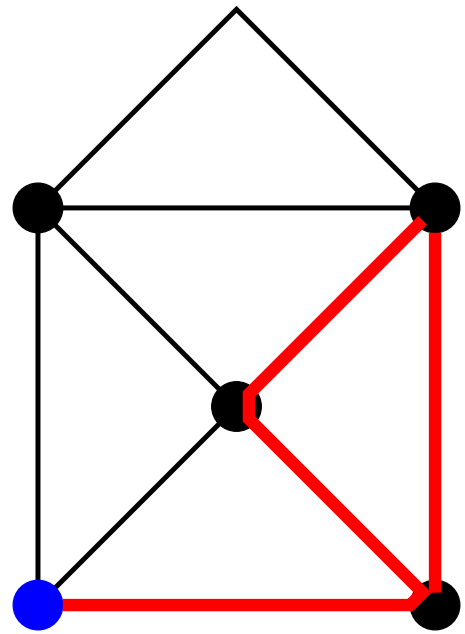
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

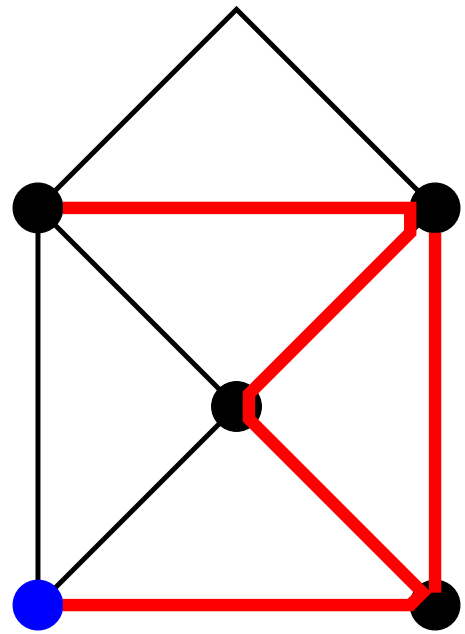
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

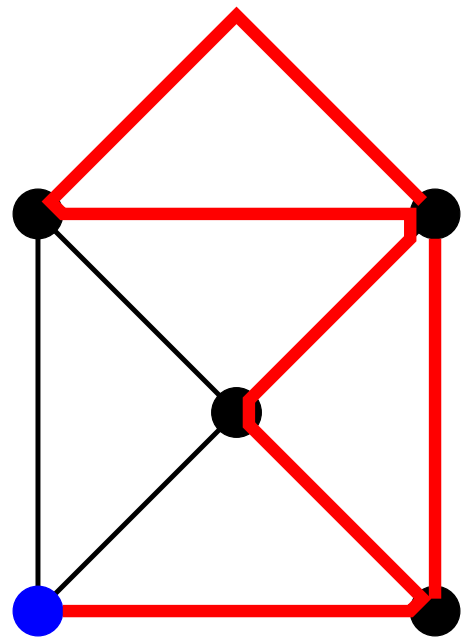
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

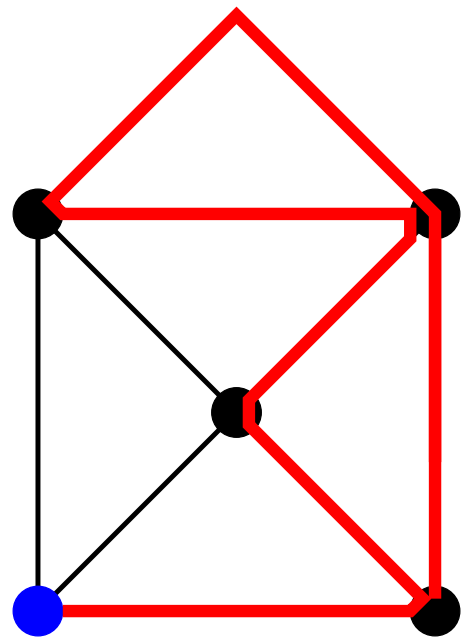
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

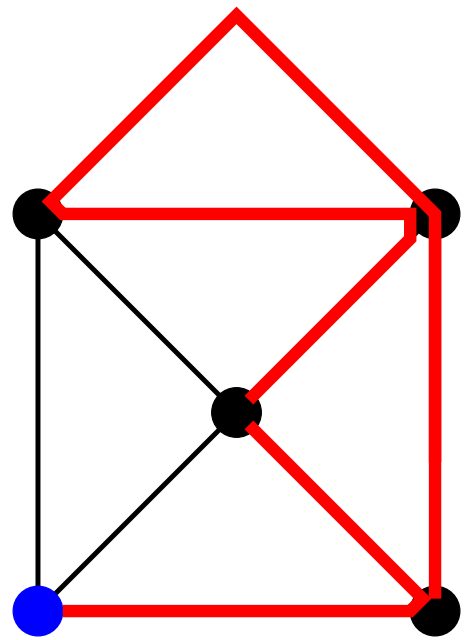
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

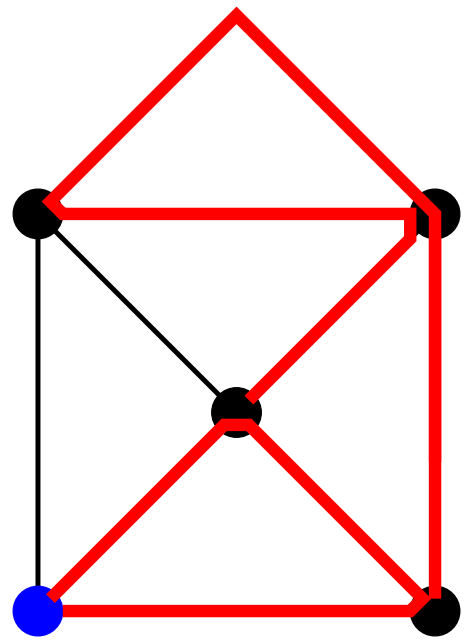
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

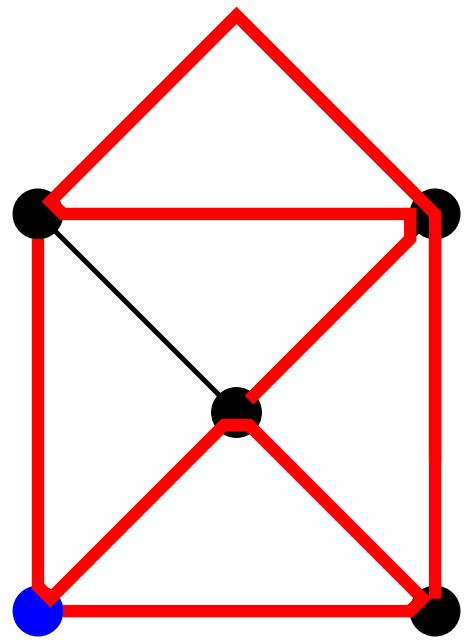
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

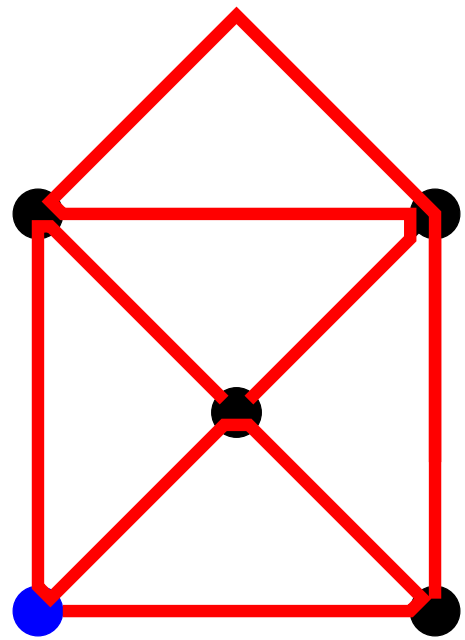
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Trouver un parcours eulérien

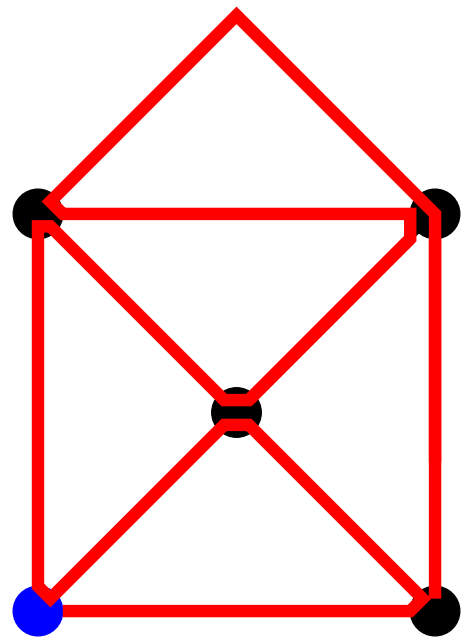
Algorithme

1. Partir d'un sommet de degré impair.
2. Avancer tant qu'on peut.
3. Si on est bloqué,

-Trouver un sommet où une arête est libre

- Couper le chemin en ce sommet.

- Repartir par une arête libre jusqu'à revenir au sommet.



Quelques applications du tour eulérien et ses généralisations

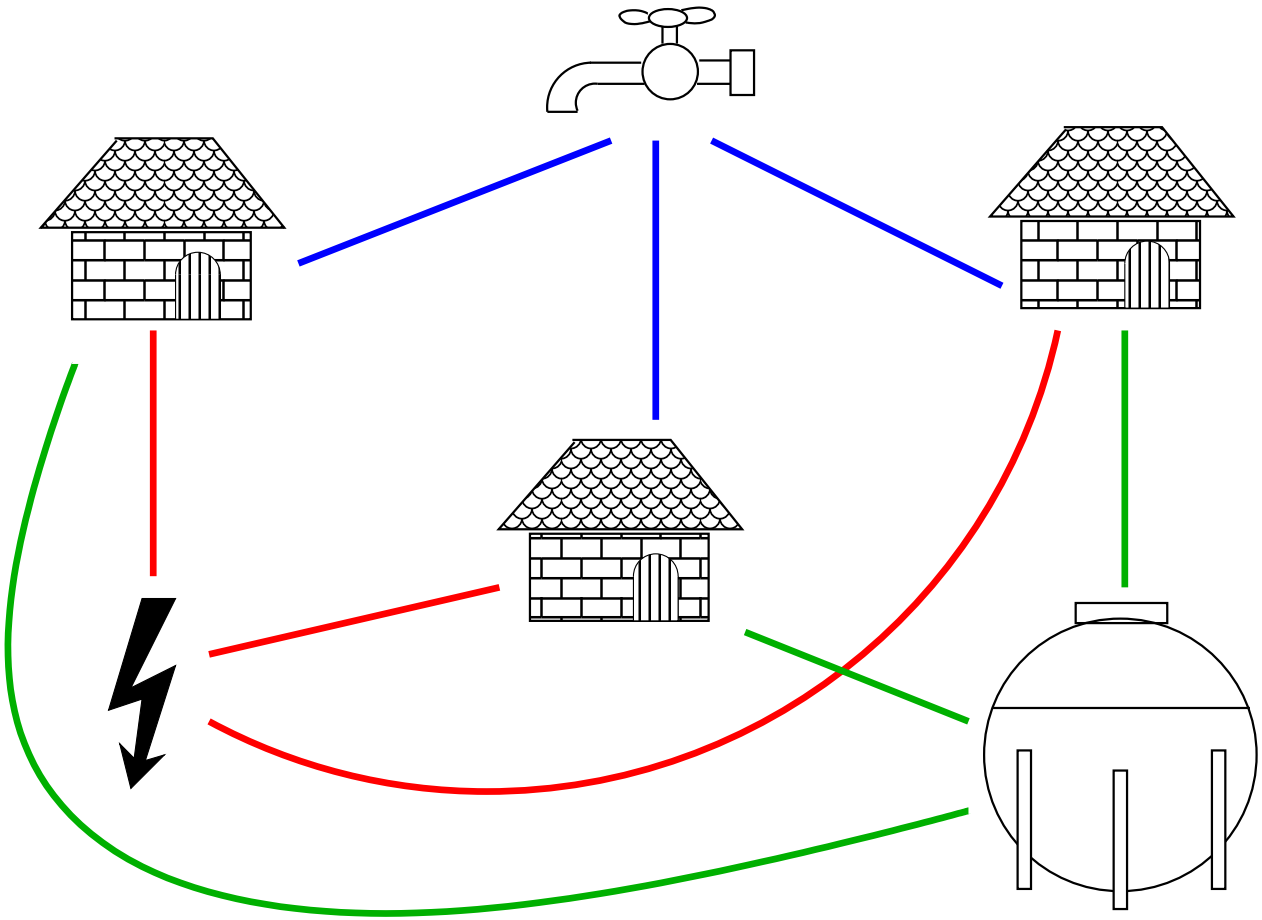
- **Tournée de camions poubelles.** Trouver un ensemble de tours $T_1, \ldots T_p$ tel que chaque rue (= arête) est dans au moins un de ces tours. On veut un tel ensemble de longueur minimum.

- **Diagnostic ou maintenance d'un réseau par un robot**.

- **Bioinformatique:** reconstruction de séquences ADN à partir de fragments.

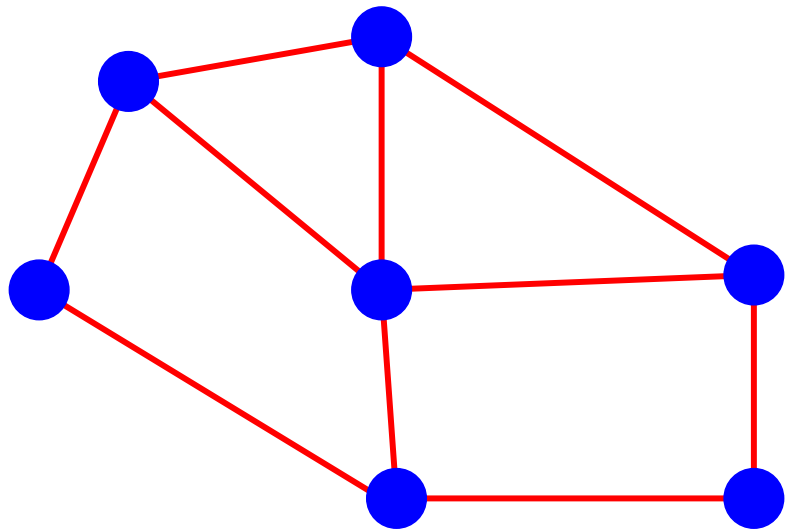
Le problème Gaz-Eau-Electricité

Peut-on relier trois maisons à l'eau, le gaz et l'électricité sans que les canalisations se croisent ?



Graphe planaire

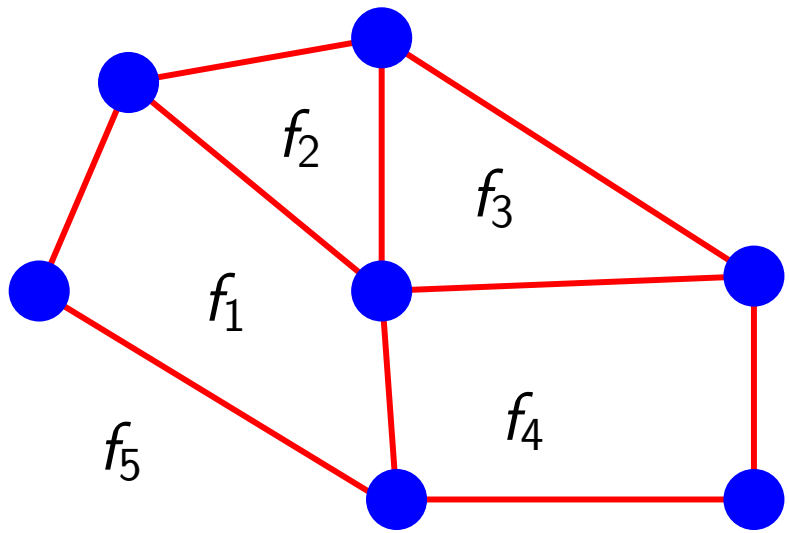
Graphe planaire: dessinable sans croisement d'arêtes.



Formule d'Euler

Euler: graphe planaire (connexe), $S + F = A + 2$

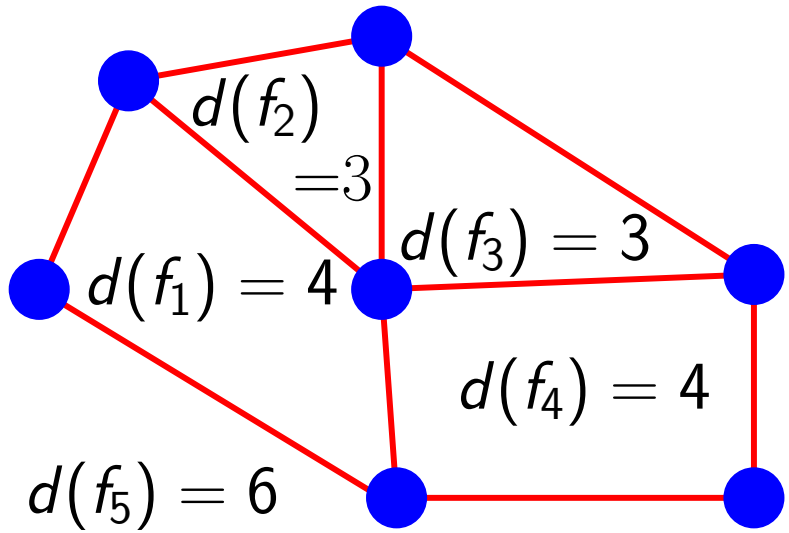
S: nombre de sommets A: nombre d'arêtes F: nombre de faces.



Ici $S = 7$, $A = 10$, $F = 5$. On a bien $7 + 5 = 10 + 2$.

Relation degré des faces – nombre d'arêtes

degré d'une face f, $d(f)$: nombre d'arêtes "autour" de f.



Une arête est autour de deux faces. Ainsi $\sum_{f \text{ face}} d(f) = 2 \times A$.

Ici: $d(f_1) + d(f_2) + d(f_3) + d(f_4) + d(f_5)$
$= 4 + 3 + 3 + 4 + 6 = 20 = 2 \times 10.$

Nombre maximum d'arêtes d'un graphe planaire

Une face a au moins 3 arêtes donc $2A = \sum_{f \text{ face}} d(f) \geq 3F$.

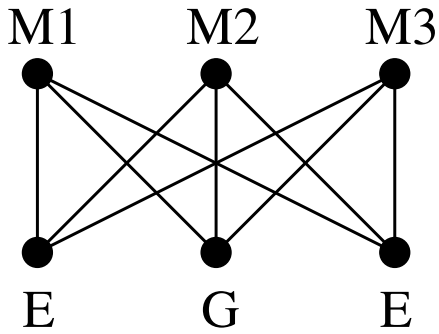
soit $F \leq \frac{2}{3}A$.

$$\begin{aligned} A &= S + F - 2 \\ &\leq S + \frac{2}{3}A - 2 \\ \frac{1}{3}A &\leq S - 2 \end{aligned}$$

D'où $A \leq 3S - 6$.

Réponse problème Gaz-Eau-Electricité

Dans notre graphe, les faces seraient de taille au moins 4.



Reprenons le calcul: $2A = \sum_{f \text{ face}} d(f) \geq 4F$, soit $F \leq \frac{1}{2}A$.

$$\begin{aligned} A &= S + F - 2 \\ &\leq S + \frac{1}{2}A - 2 \\ \frac{1}{2}A &\leq S - 2 \end{aligned}$$

D'où $A \leq 2S - 4$.

Or le graphe à réaliser a 6 sommets et 9 arêtes. Il ne peut pas être planaire.

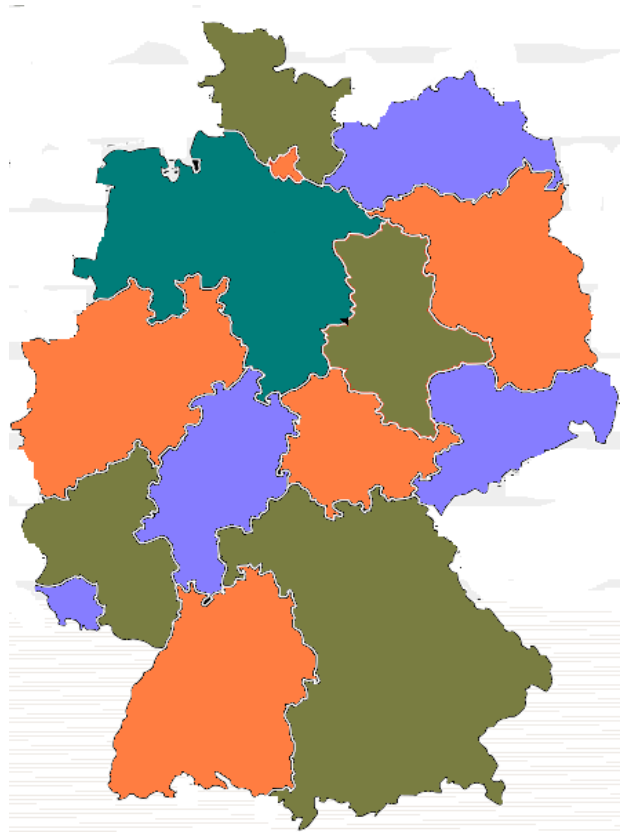
Application des graphes planaires

- **Conception de circuits intégrés** (sans croisement).
  Combinaison de milliers de transistors sur une puce.

- **Planification de réseaux**: routiers, de télécommunications, ...

Colorer une carte

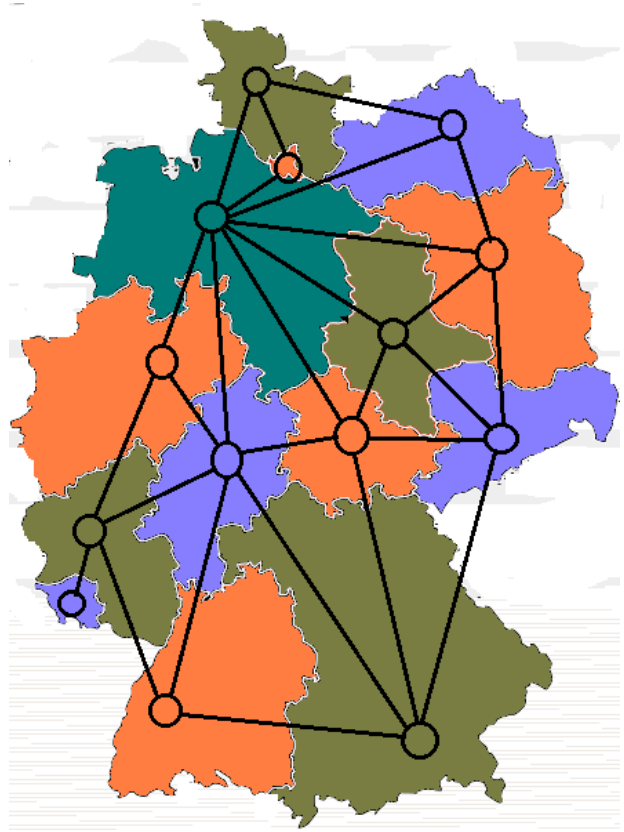
1852 Francis Guthrie: Peut-on colorer les régions (connexes) d'une carte avec 4 couleurs de manière à ce que deux régions voisines (ayant une frontière en commun) aient des couleurs différentes.



Colorer un graphe planaire

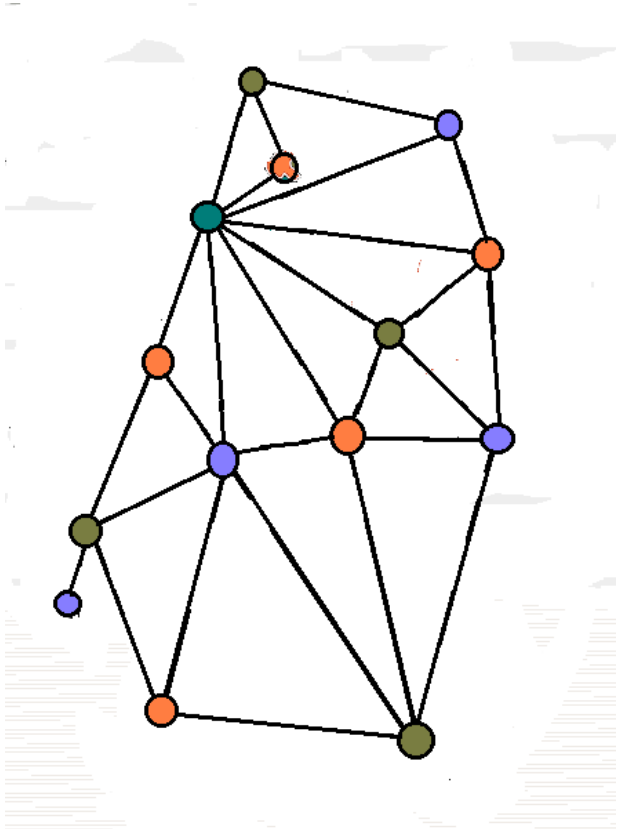
Un sommet dans chaque région.
Deux sommets reliés si les régions sont voisines.

On obtient un graphe planaire.



Colorer un graphe planaire

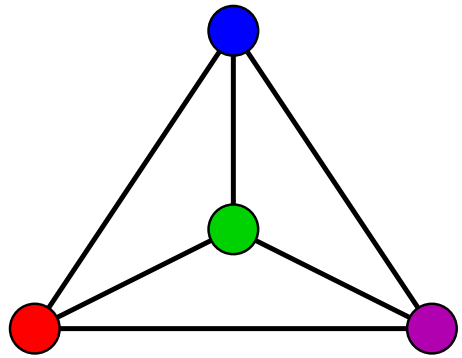
Colorer les régions = colorer le graphe planaire.
Donner des couleurs aux sommets tels que deux sommets adjacents aient des couleurs différentes.



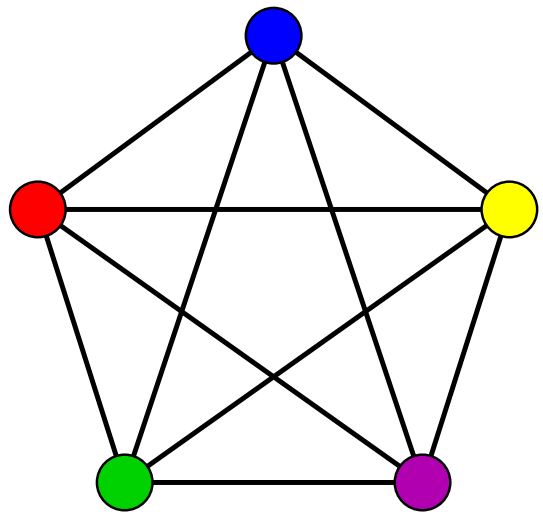
Colorer un graphe planaire

Guthrie: Peut-on colorer n'importe quel graphe planaire avec 4 couleurs?

4 couleurs nécessaires pour certains graphes.



Graphe complet à 5 sommets n'est pas planaire.
Nb d'arêtes: $(5 \times 4)/2 = 10 > 9 = 3 \times 5 - 6$



Colorer un graphe planaire avec 6 couleurs

$A \leq 3S - 6$ et $\sum_{v \text{ sommet}} d(v) = 2A$.

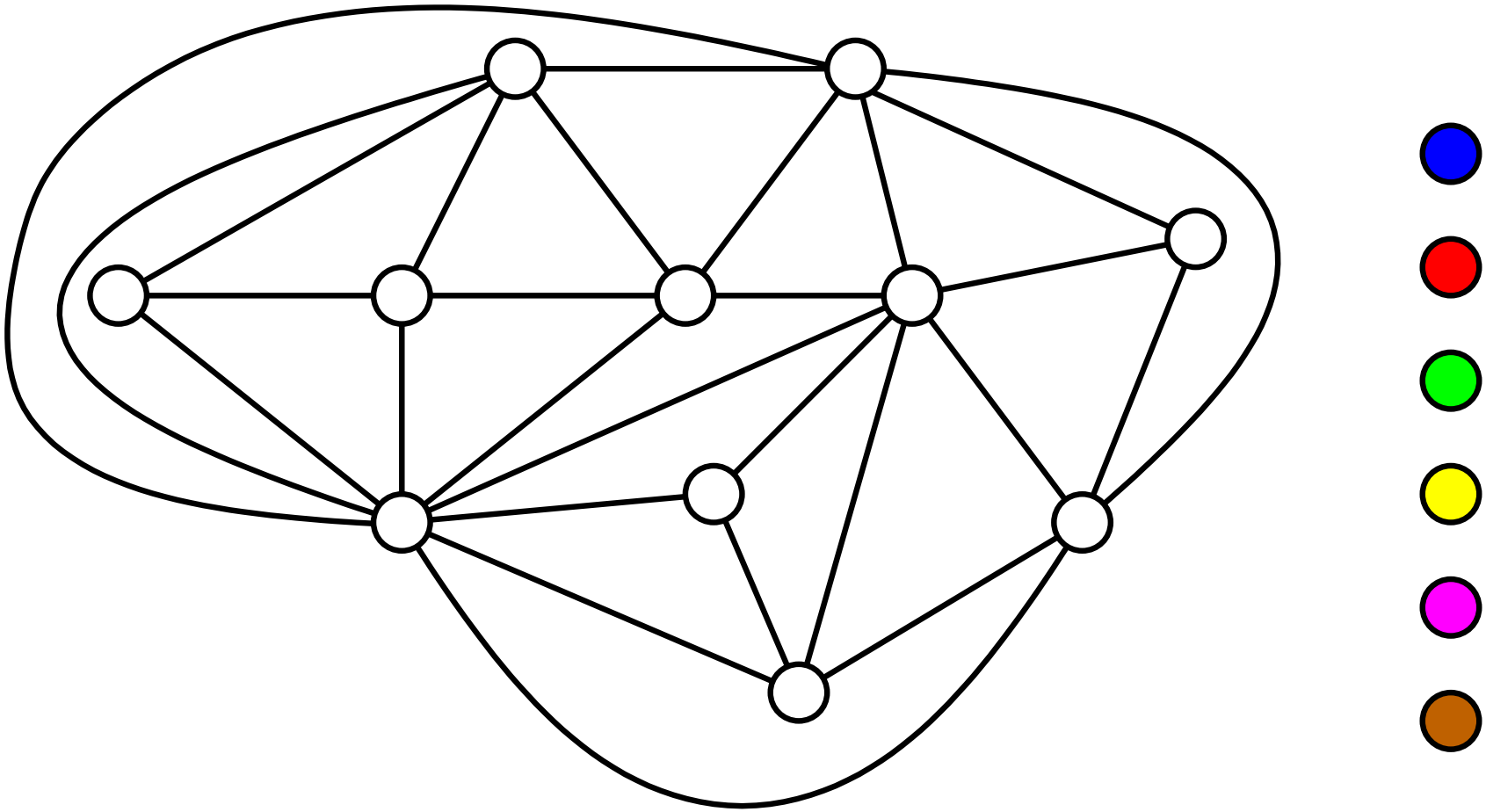
Donc $\sum_{v \text{ sommet}} d(v) \leq 6S - 12 < 6S$.

Il existe v tel que $d(v) < 6$, soit $d(v) \leq 5$.

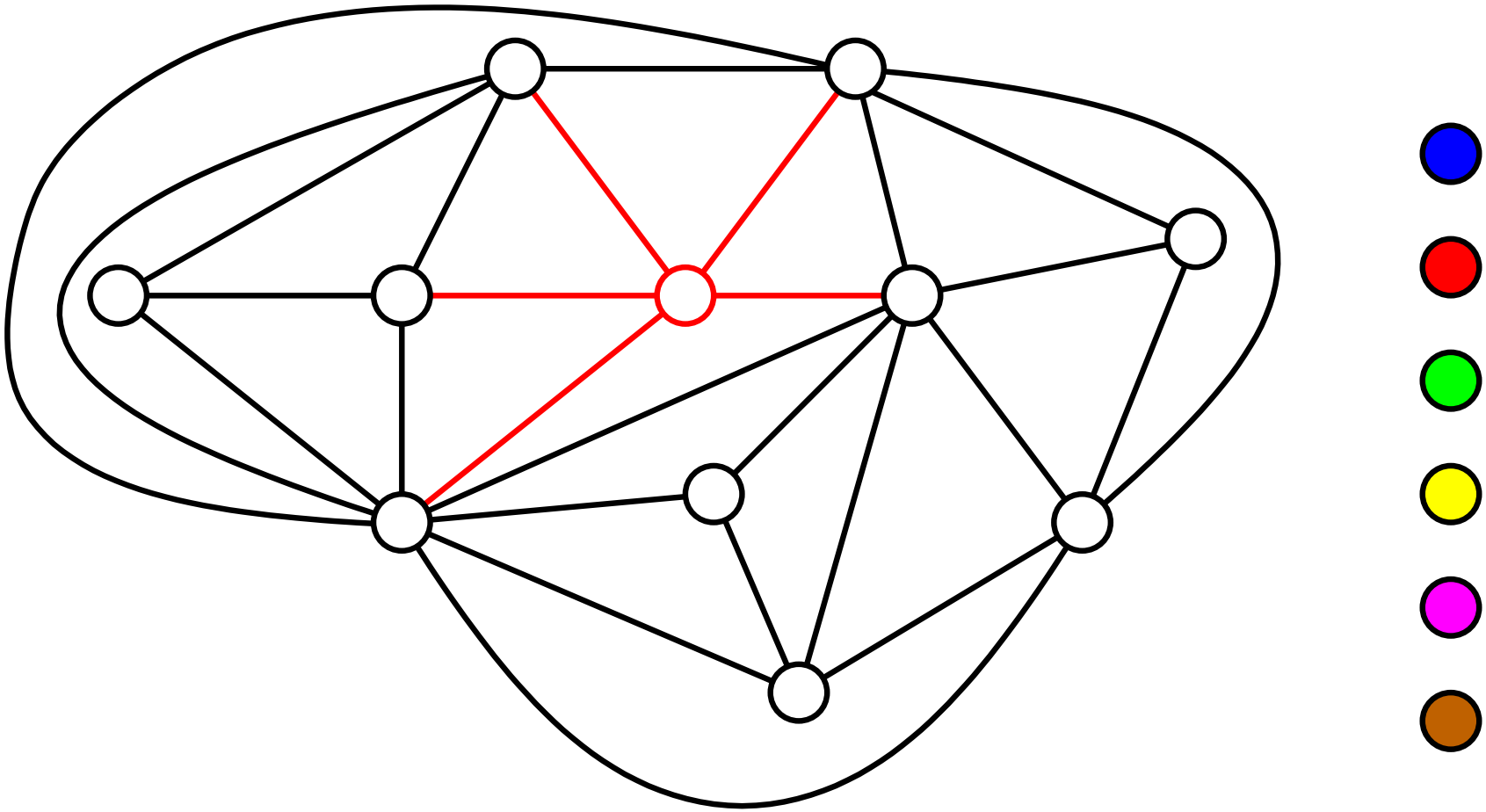
Ordre $(v_1, \ldots, v_n)$ des sommets t.q. v_i a au plus 5 voisins dans $\{v_1, \ldots, v_{i-1}\}$.

En colorant suivant cet ordre, on utilise au plus 6 couleurs.

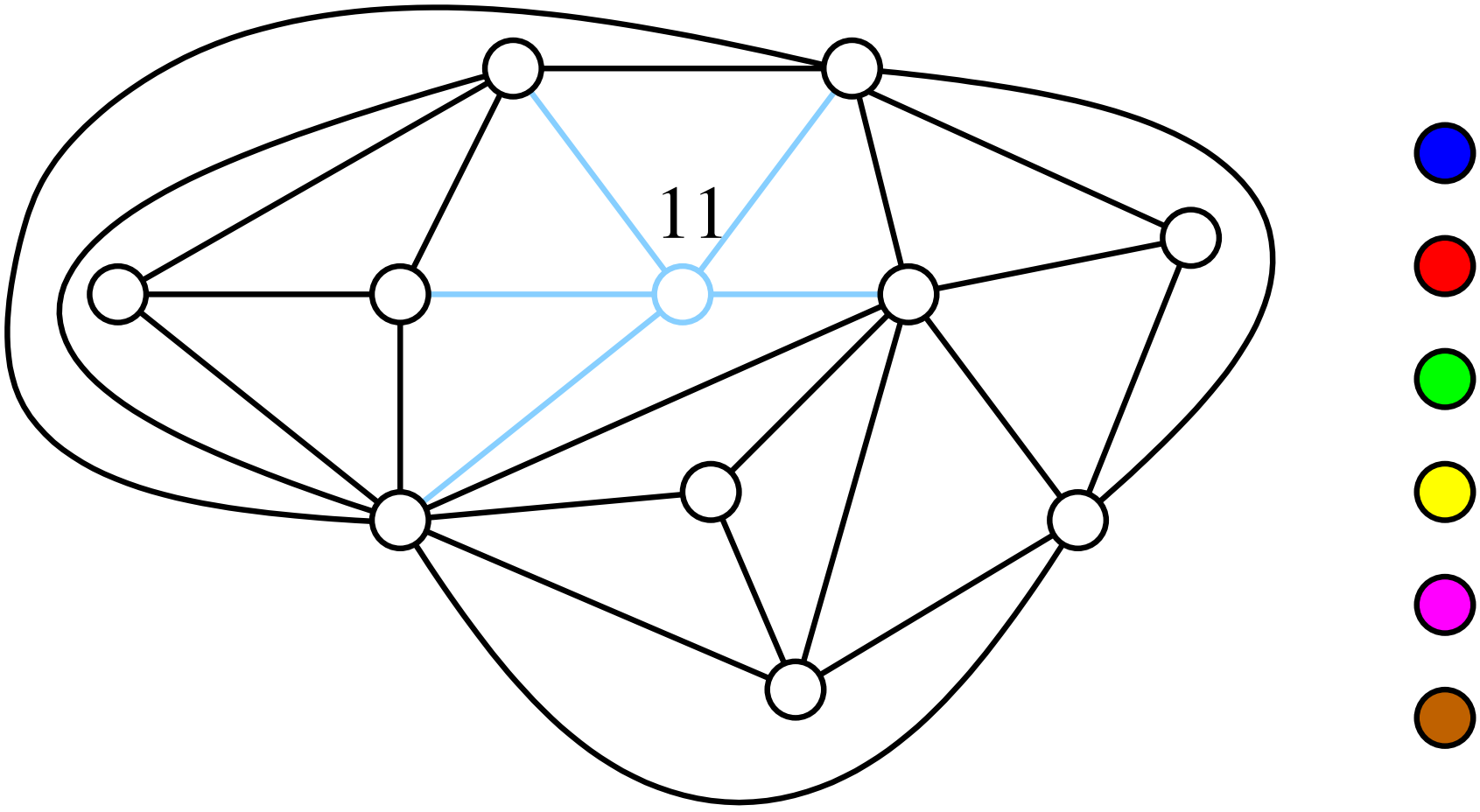
Exemple



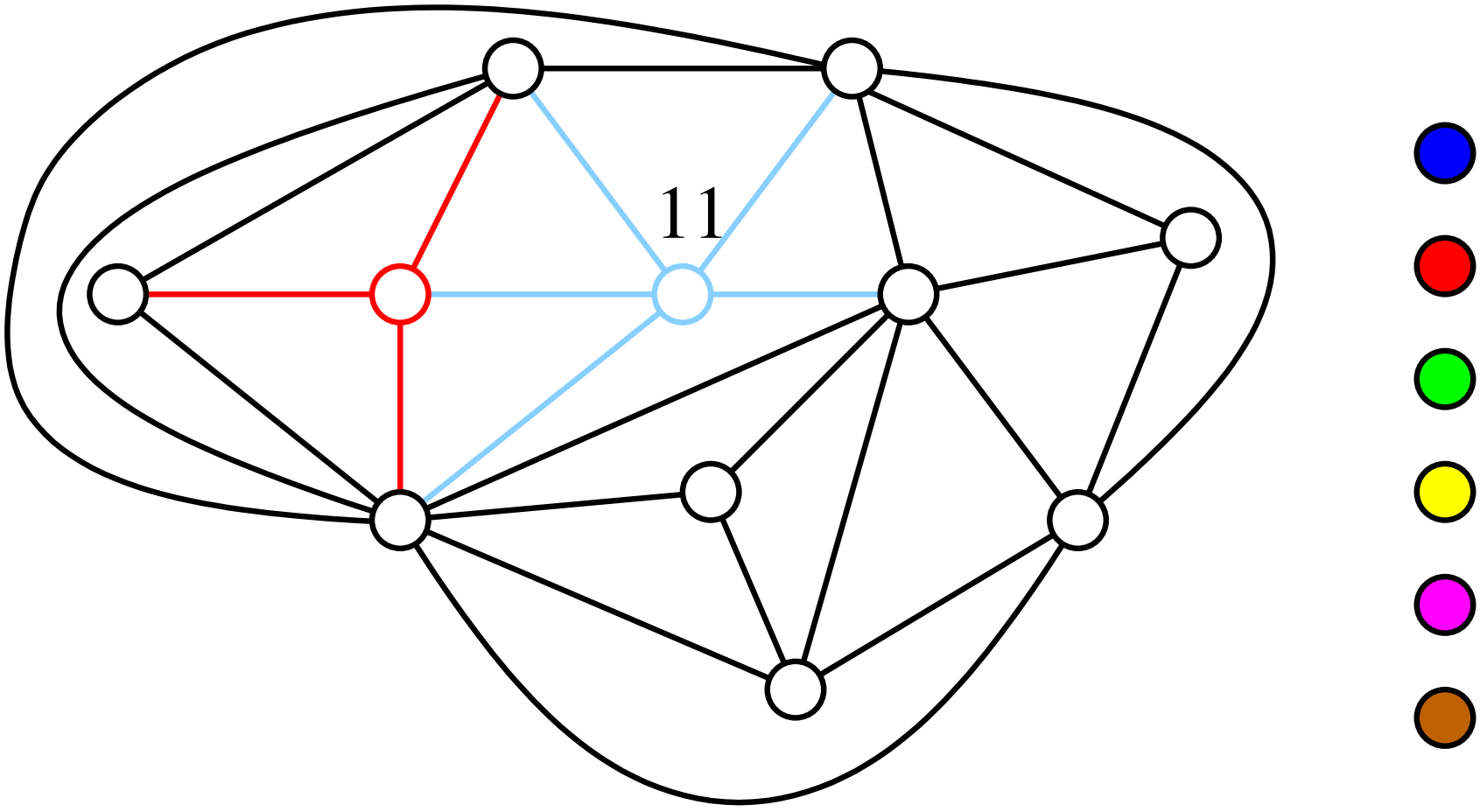
Exemple



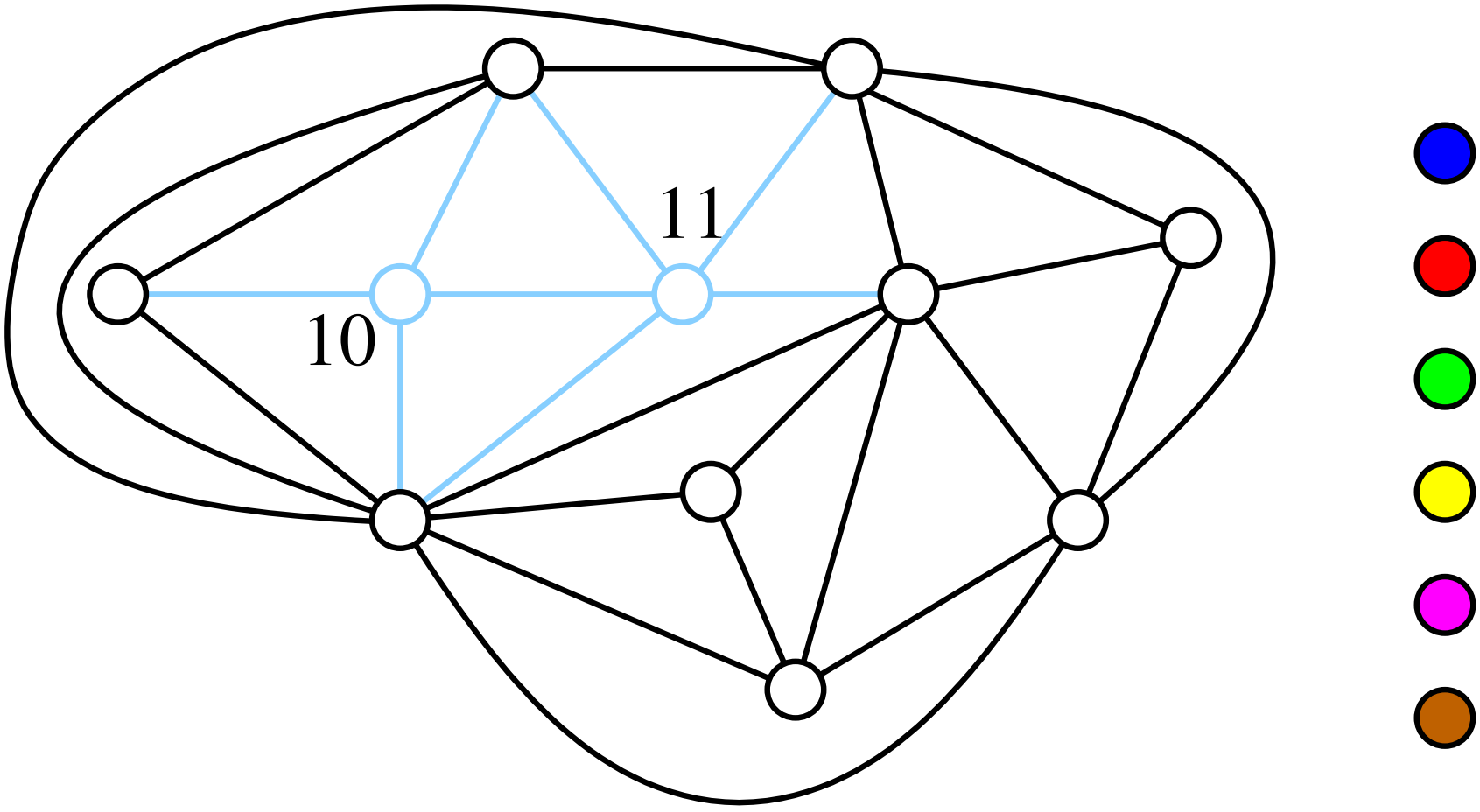
Exemple



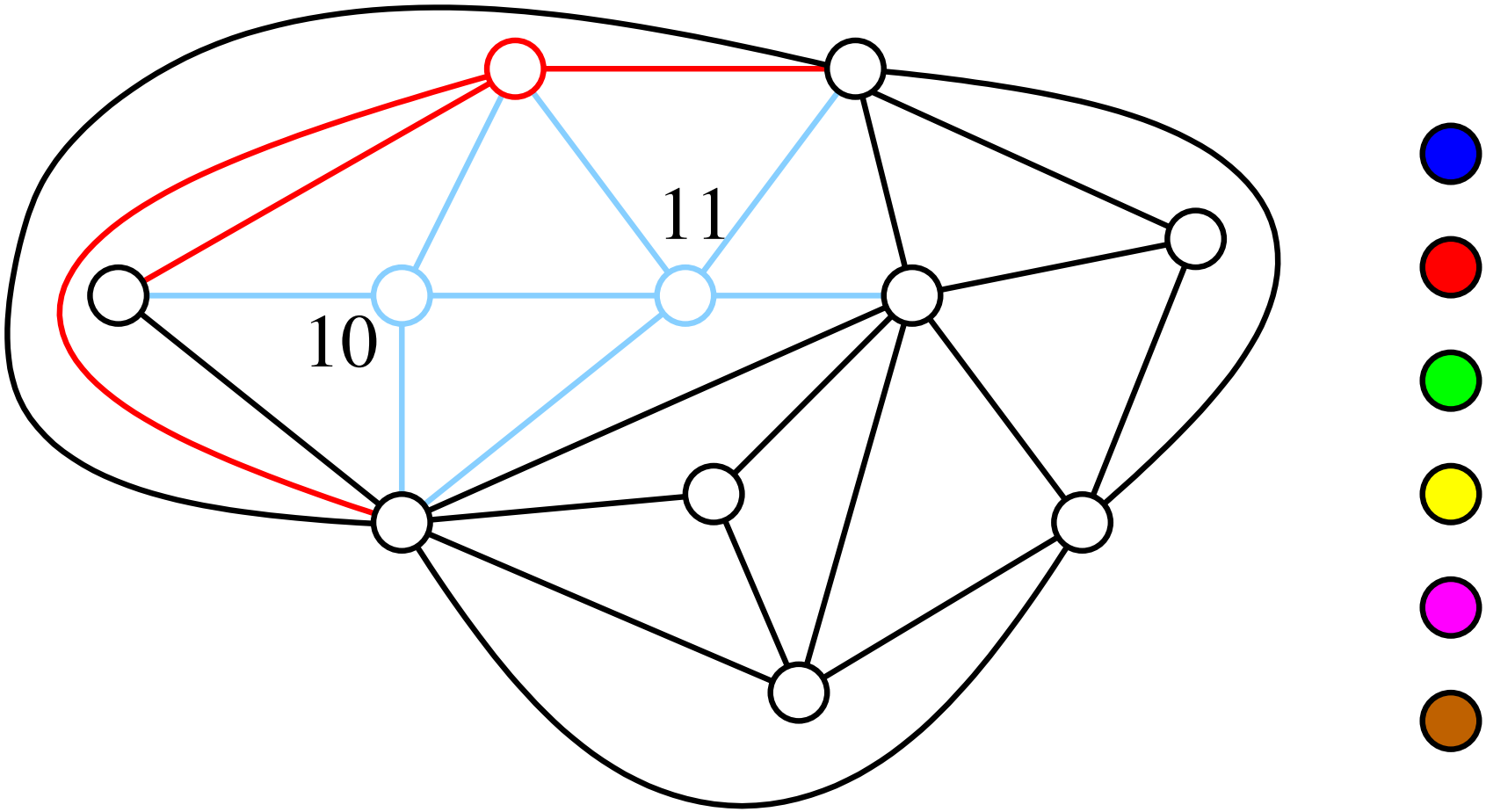
Exemple



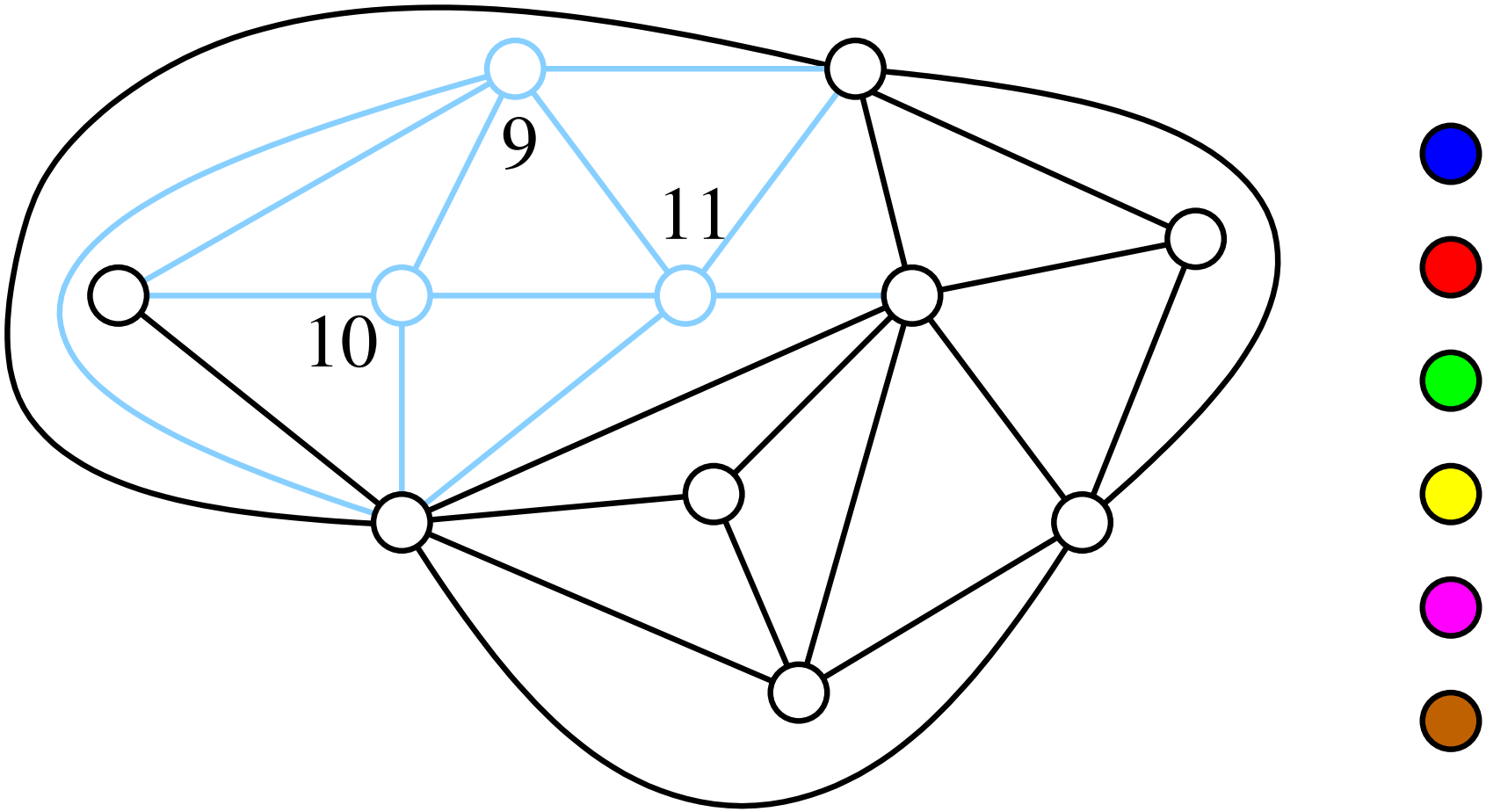
Exemple



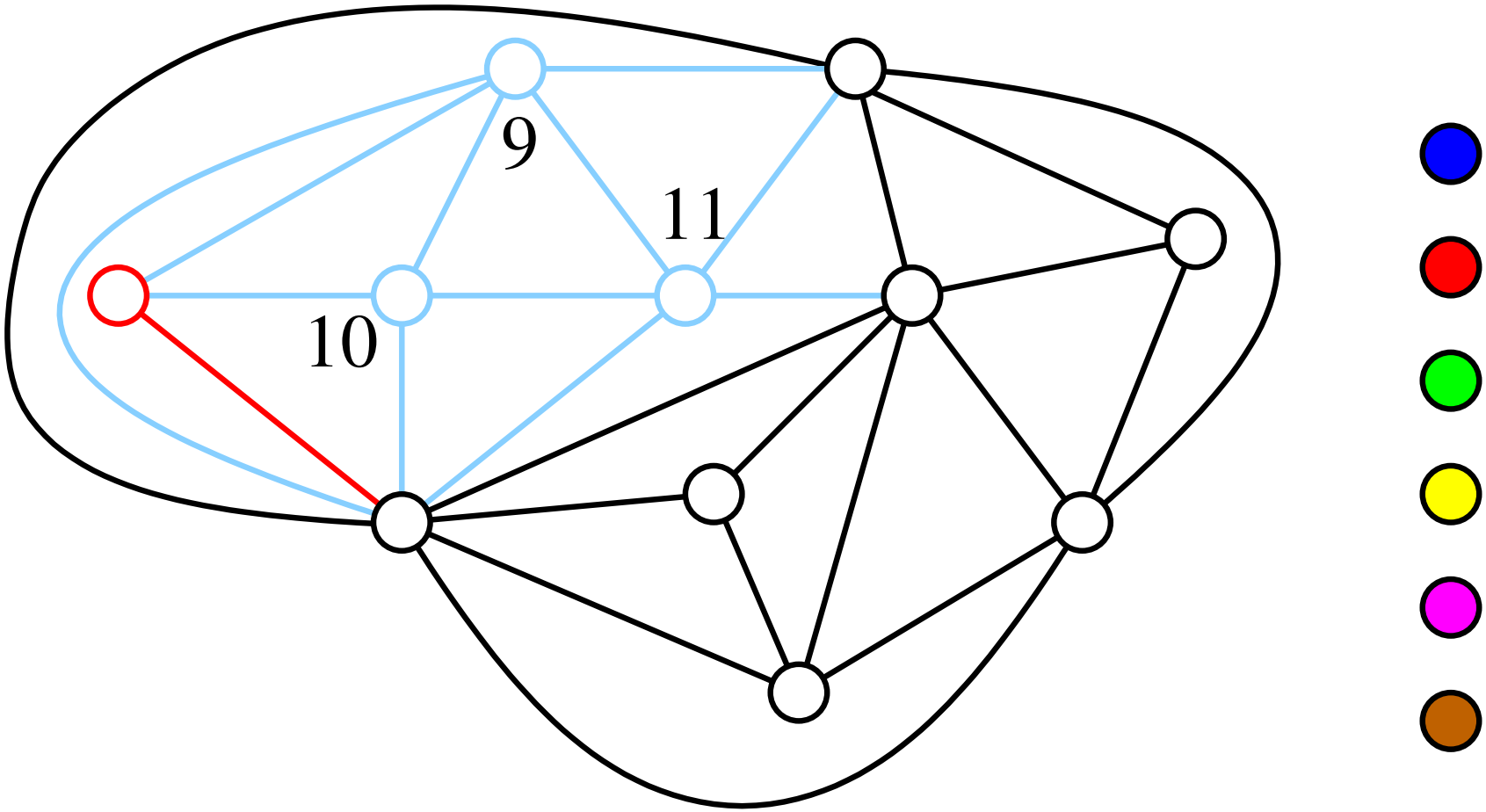
Exemple



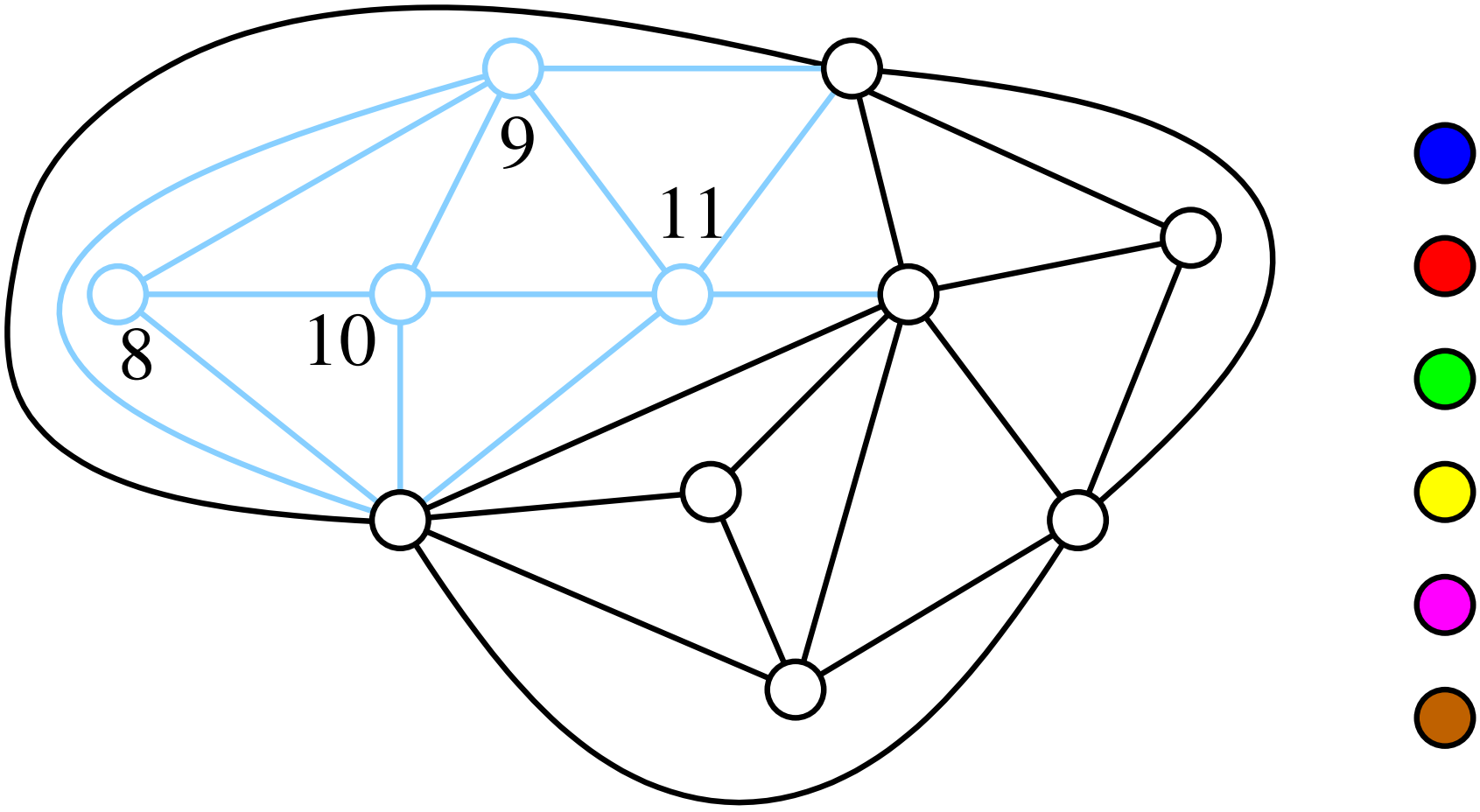
Exemple



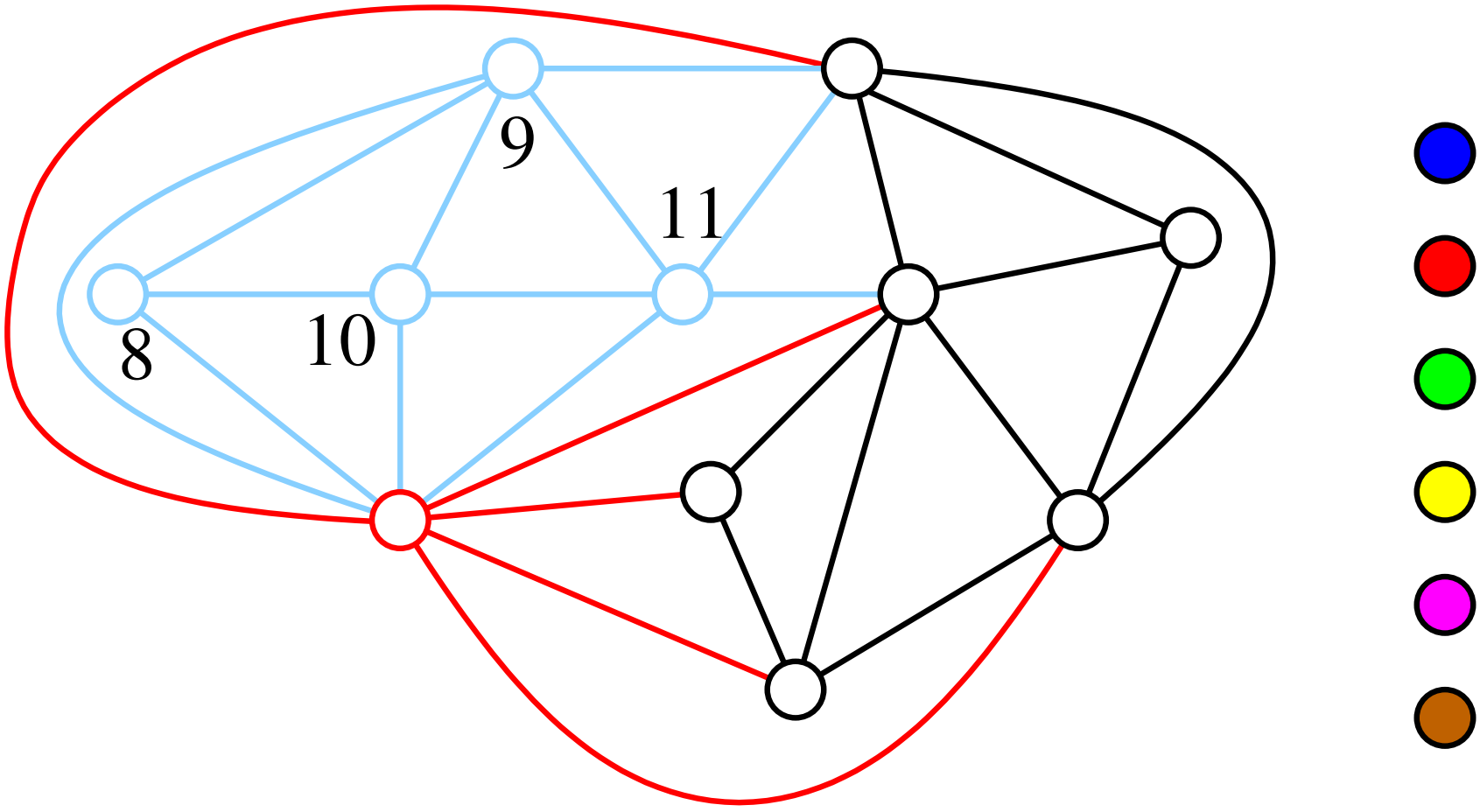
Exemple



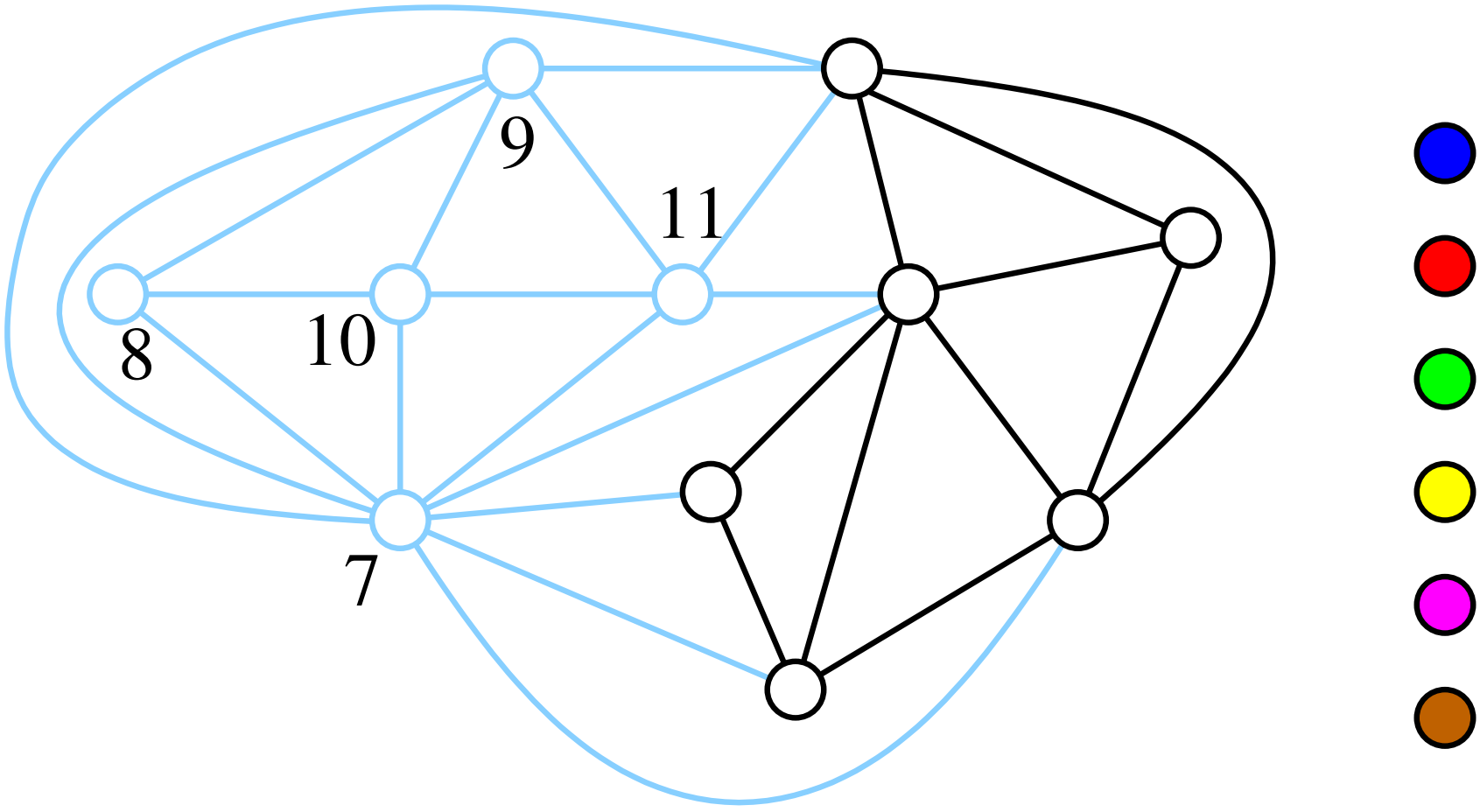
Exemple



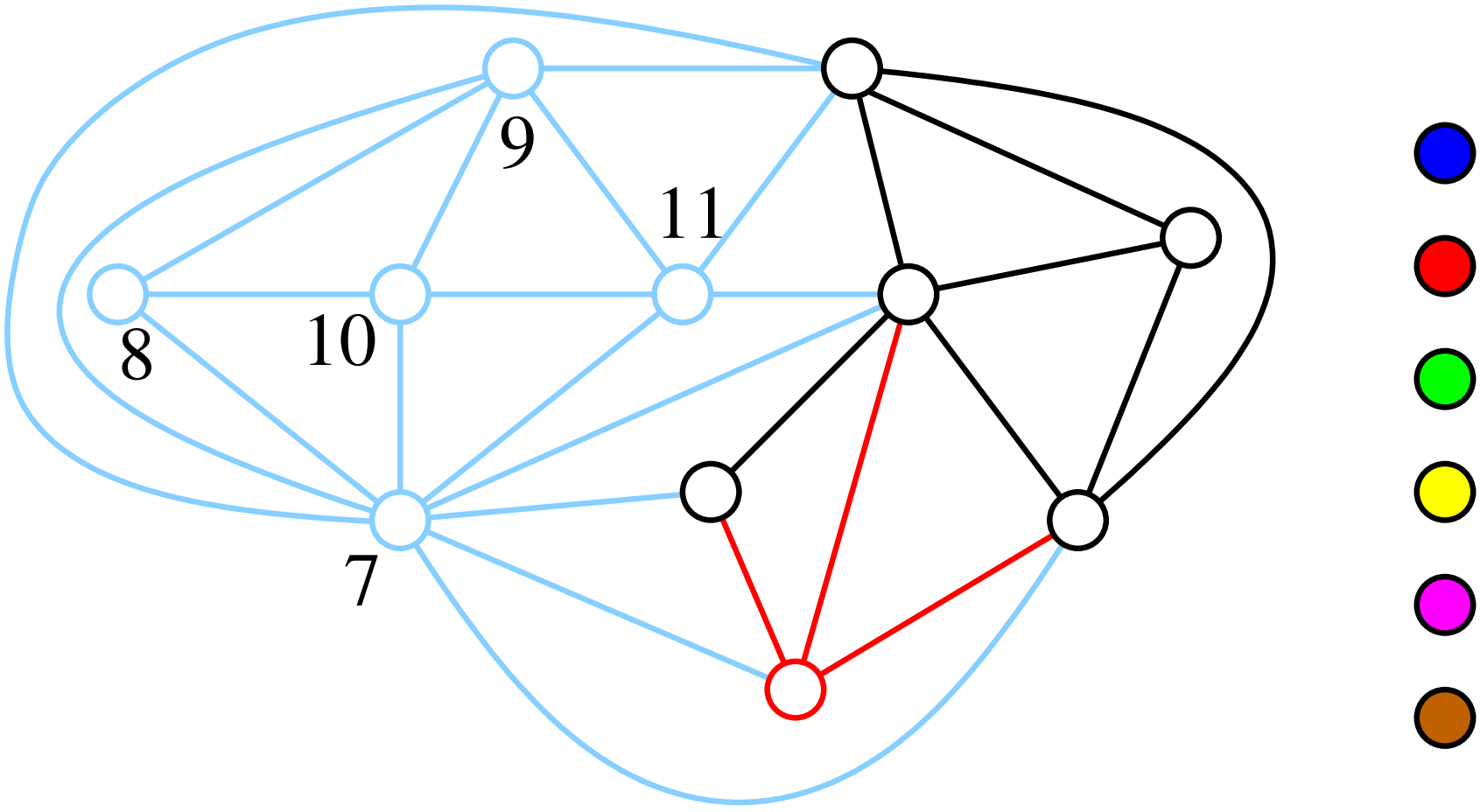
Exemple



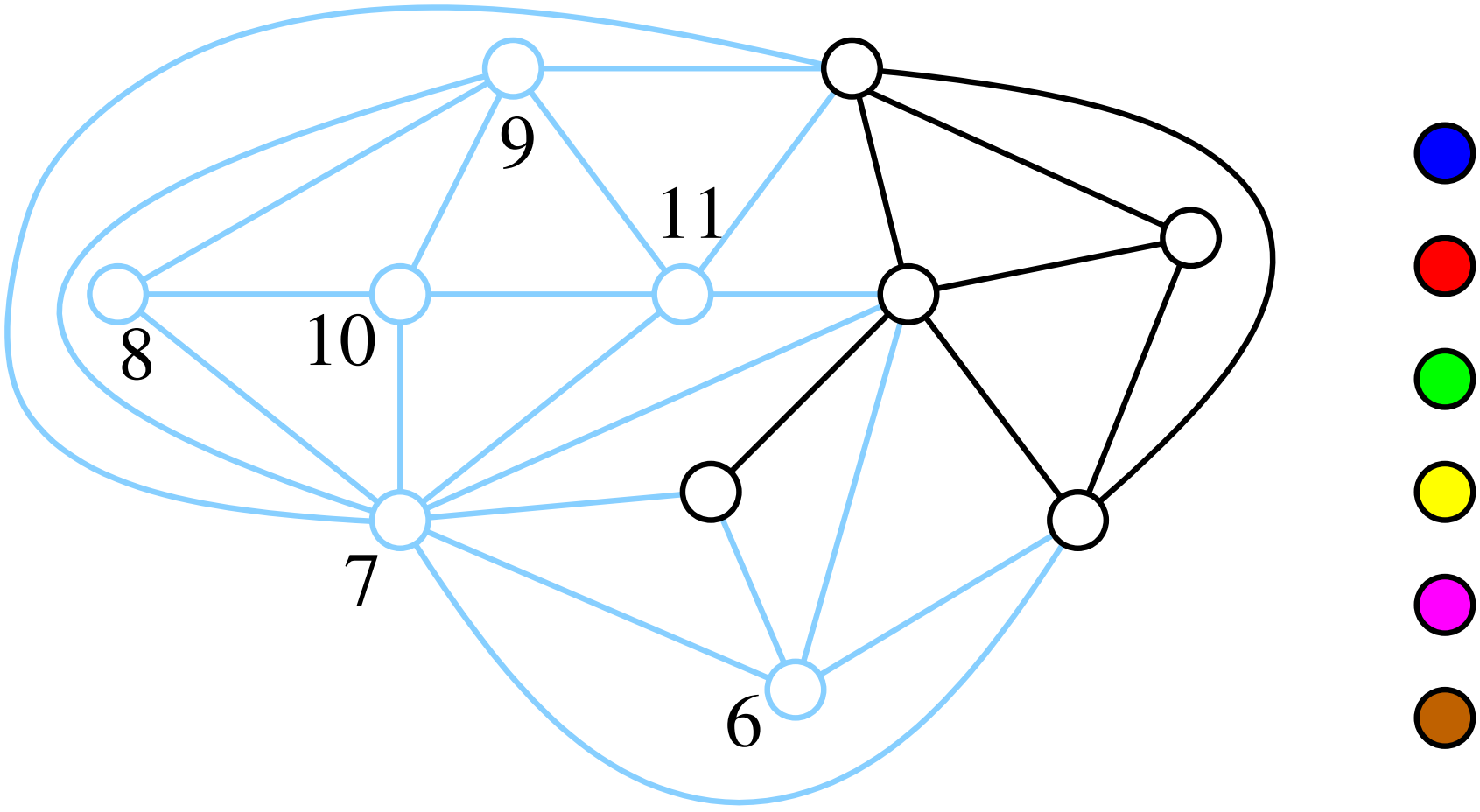
Exemple



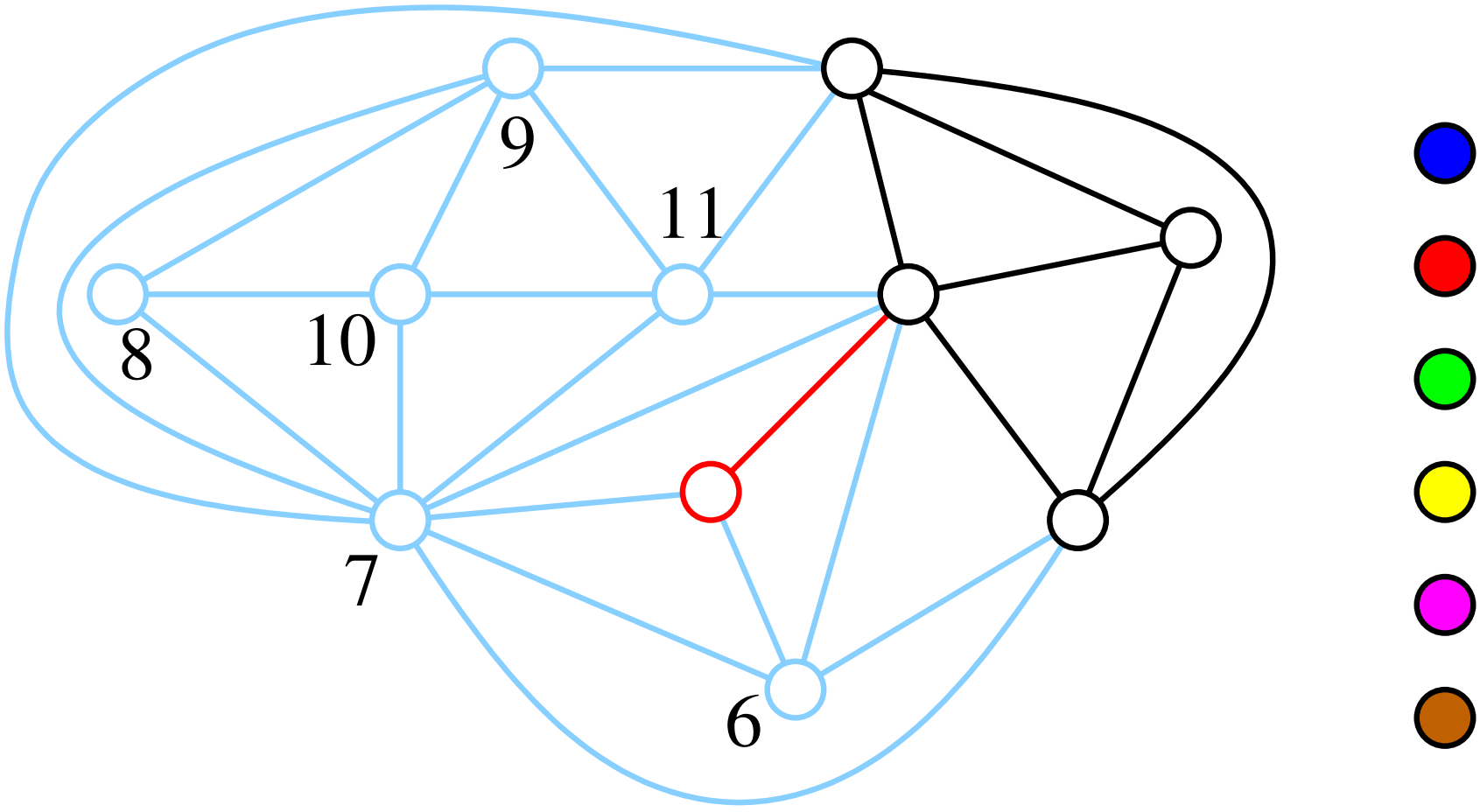
Exemple



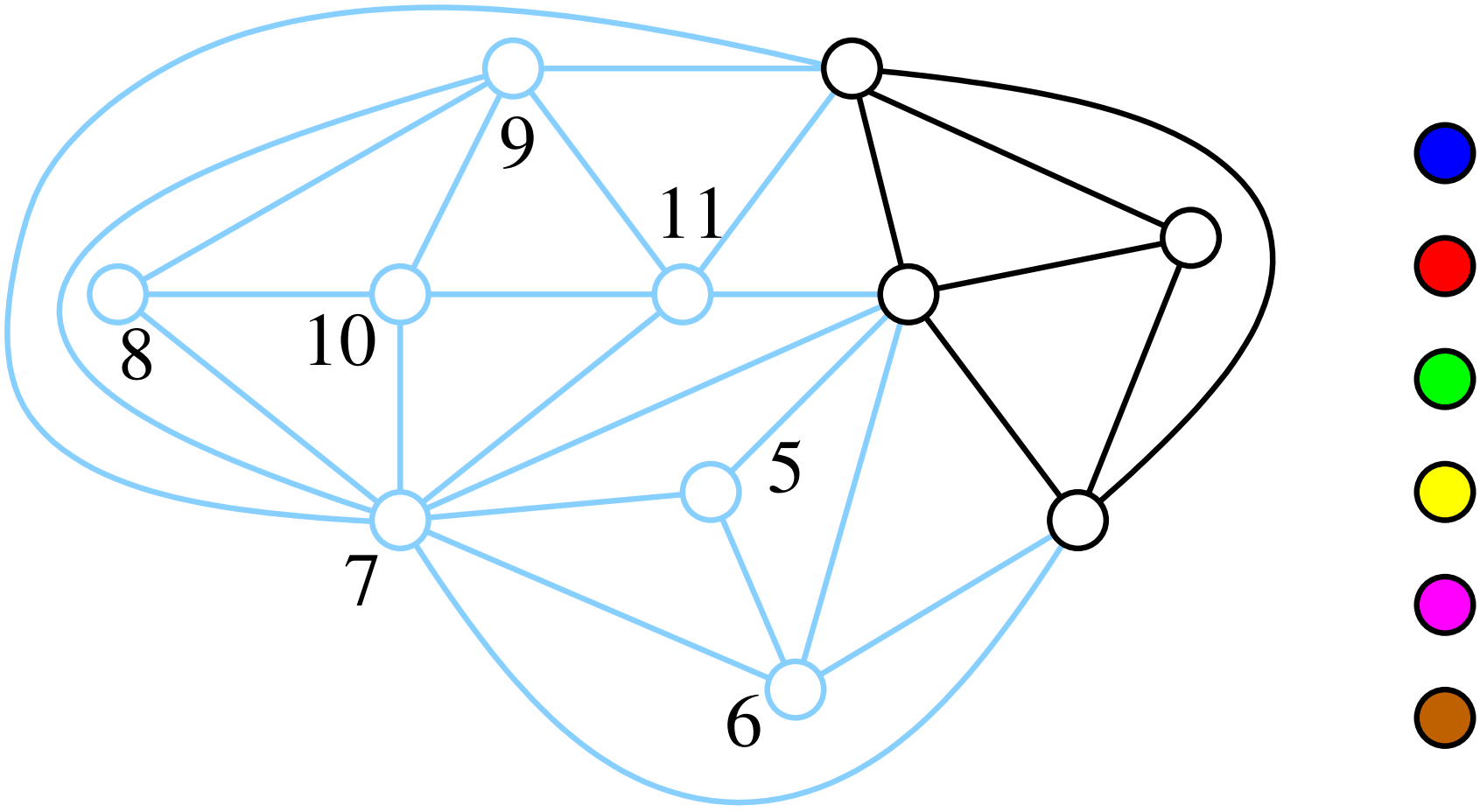
Exemple



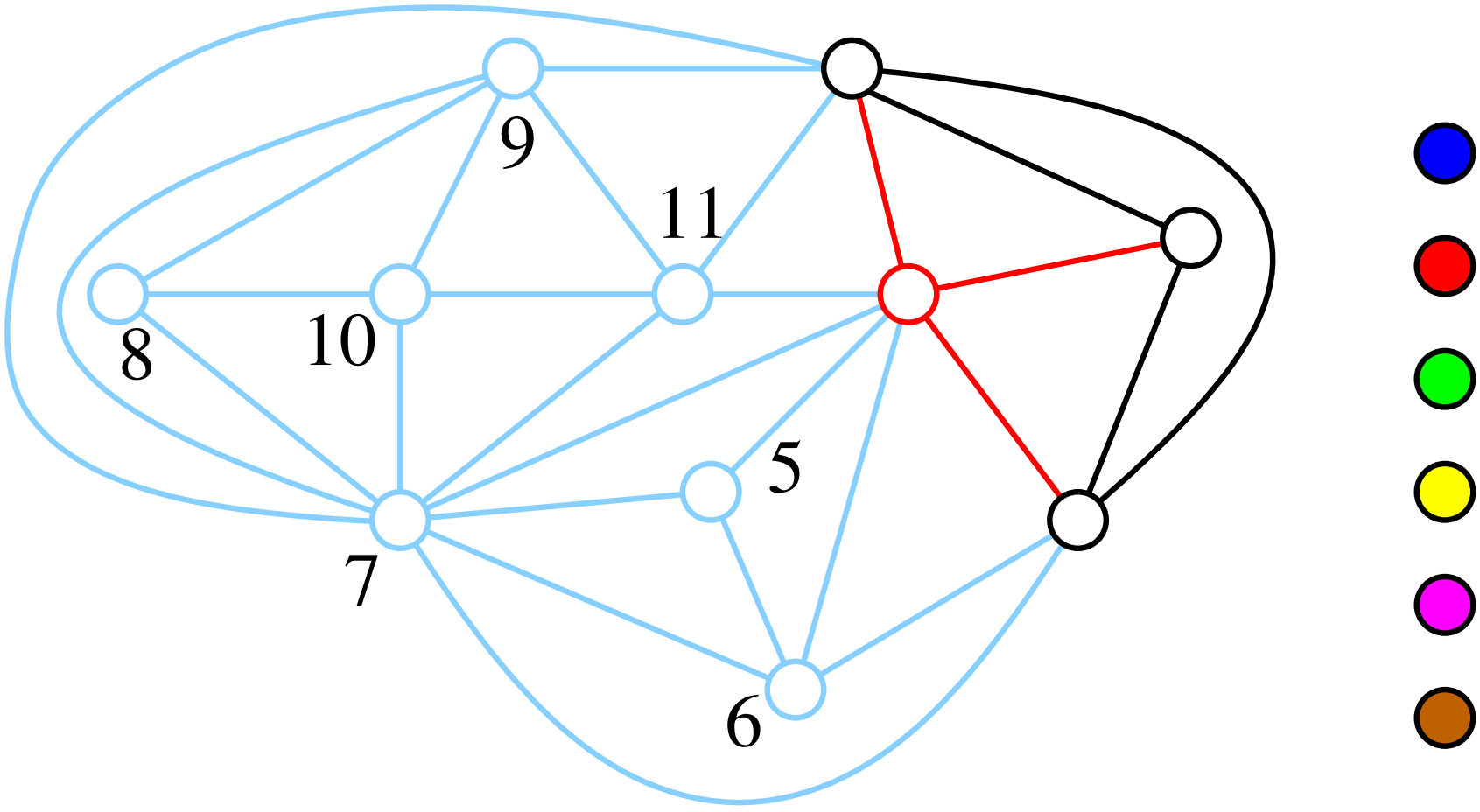
Exemple



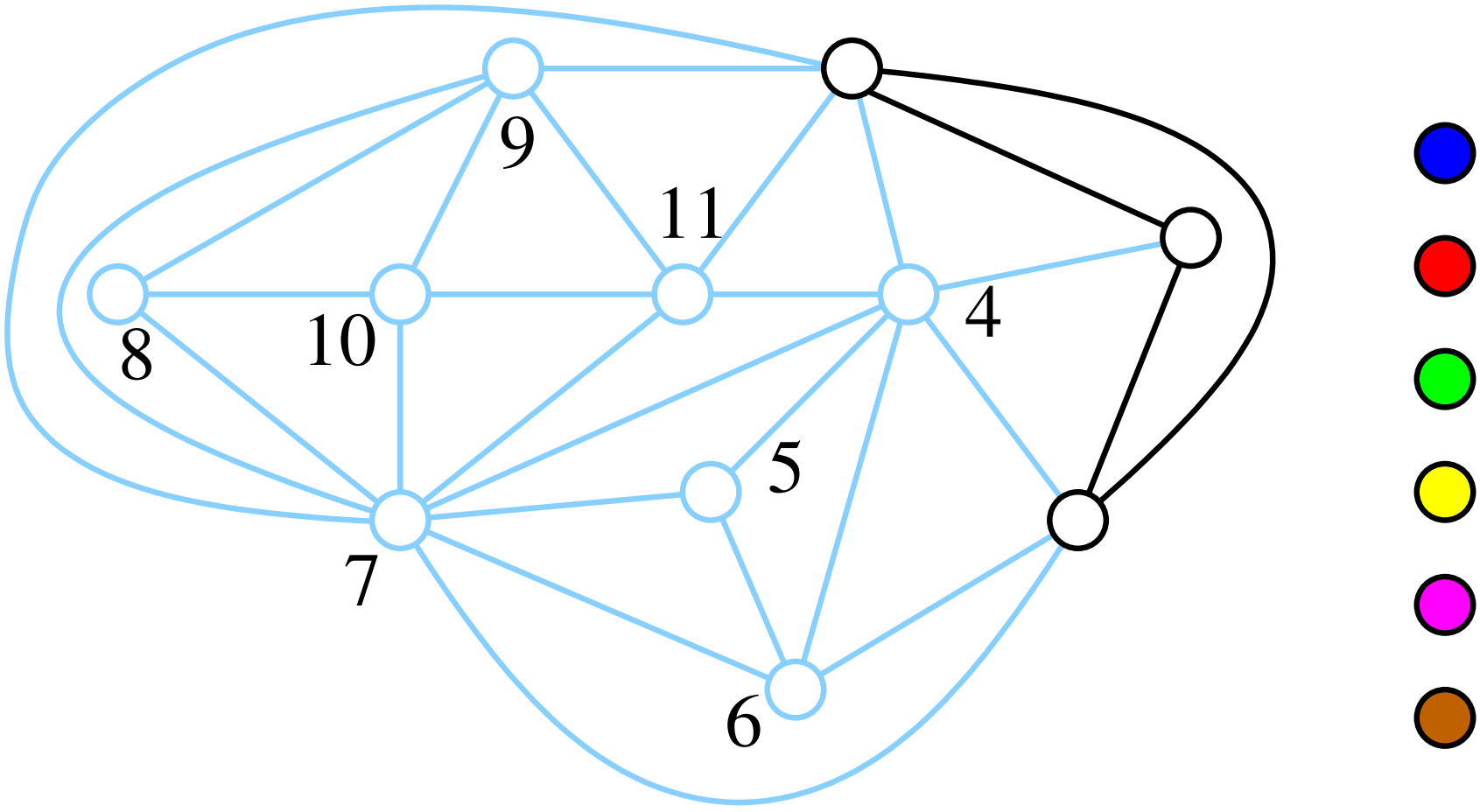
Exemple



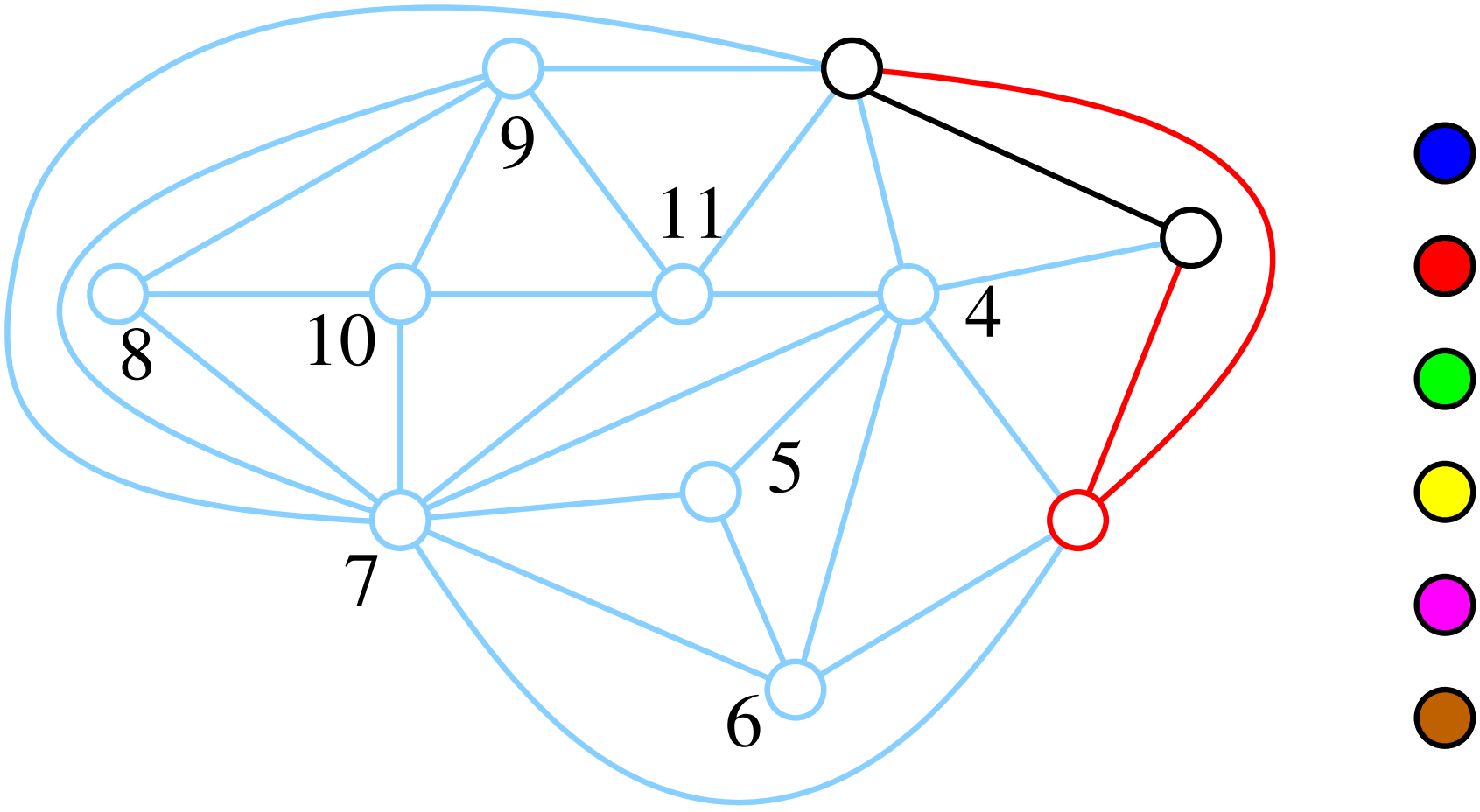
Exemple



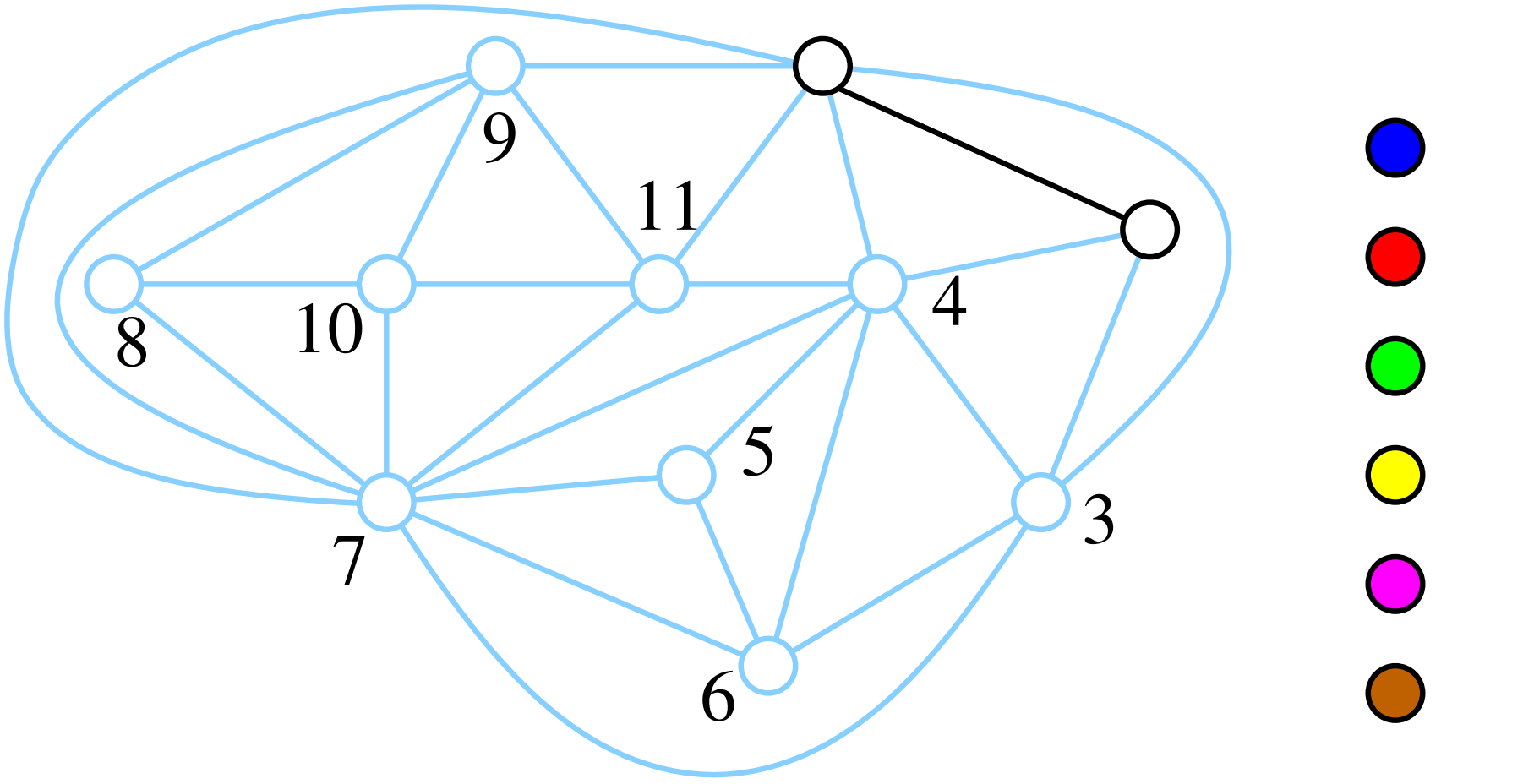
Exemple



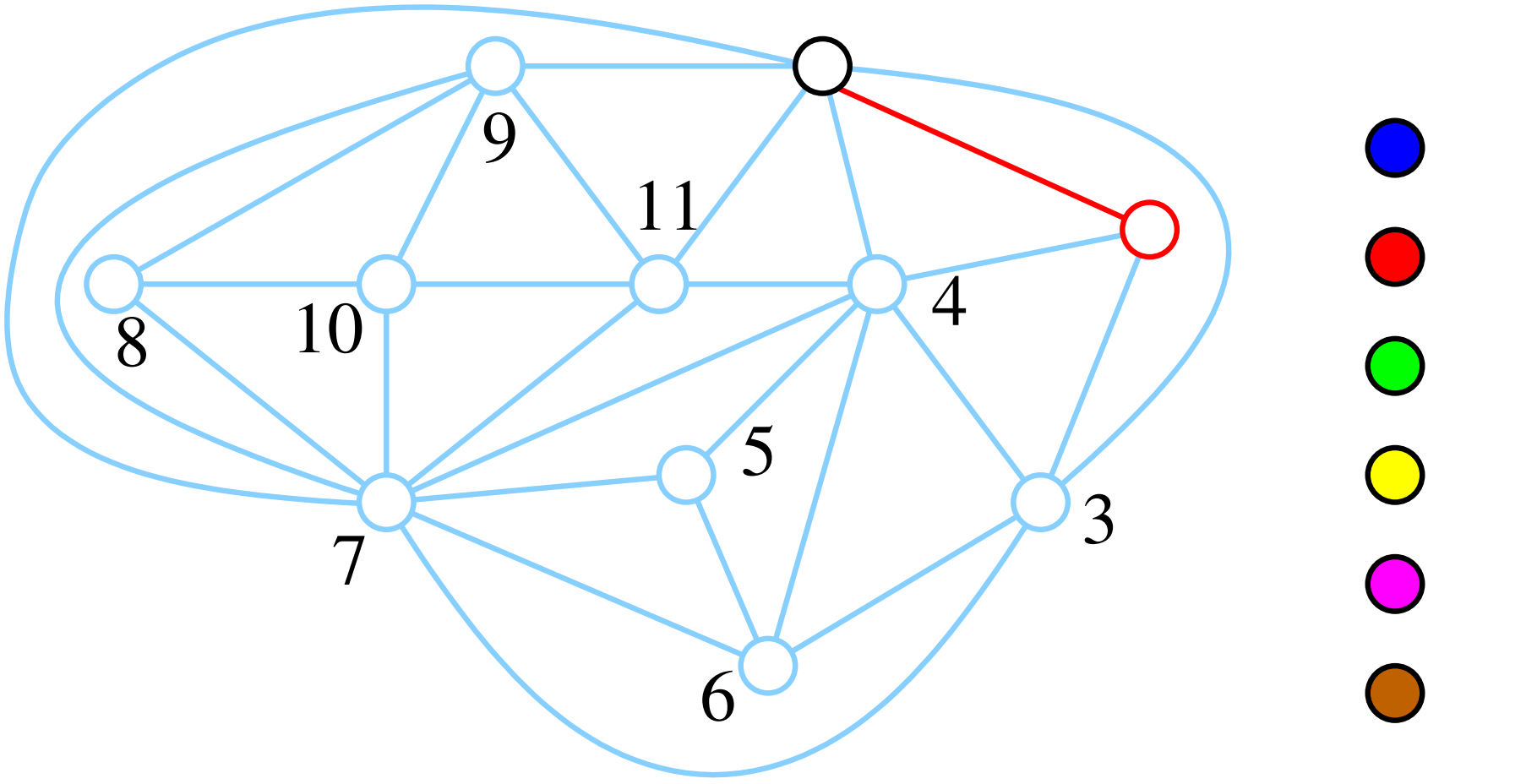
Exemple



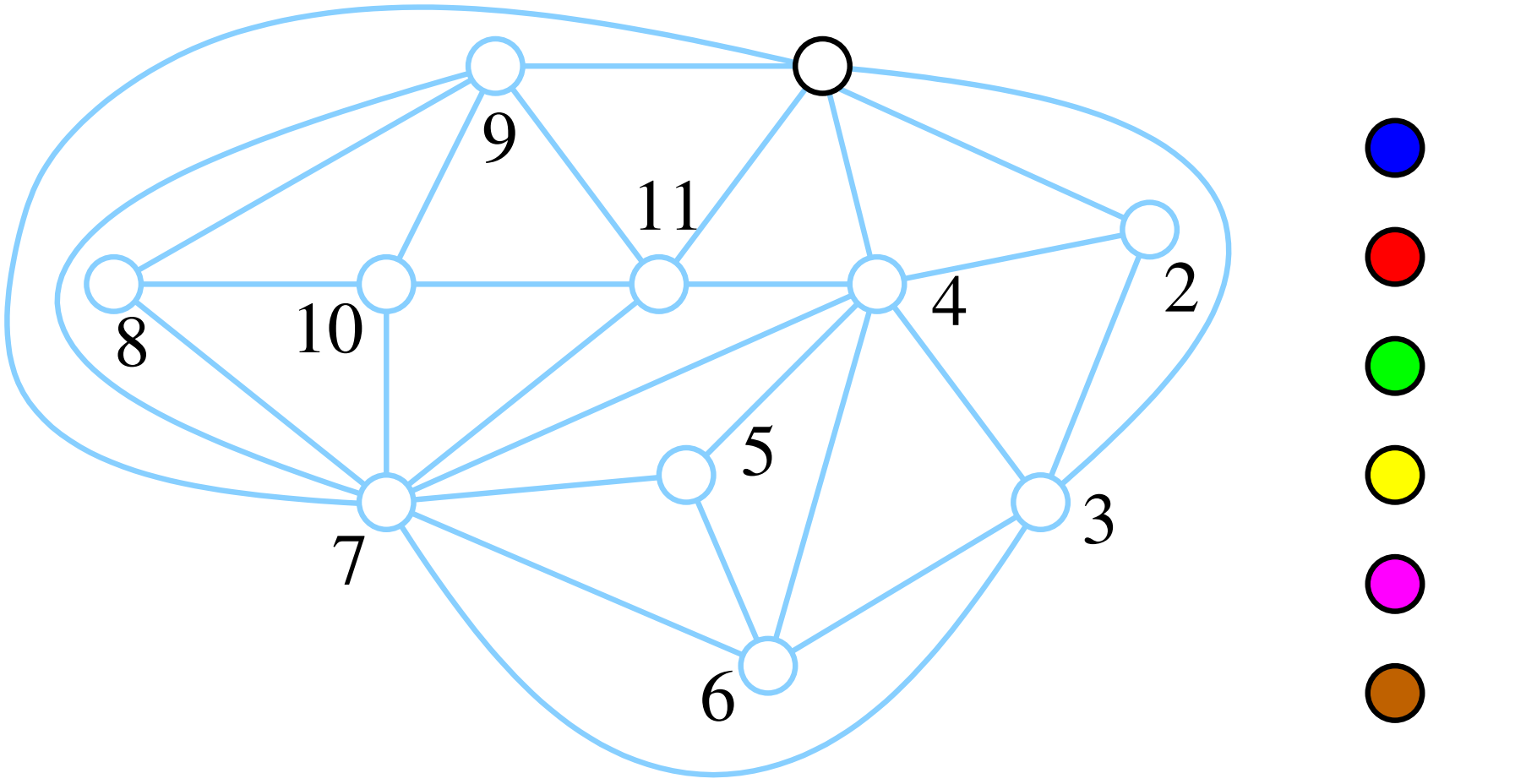
Exemple



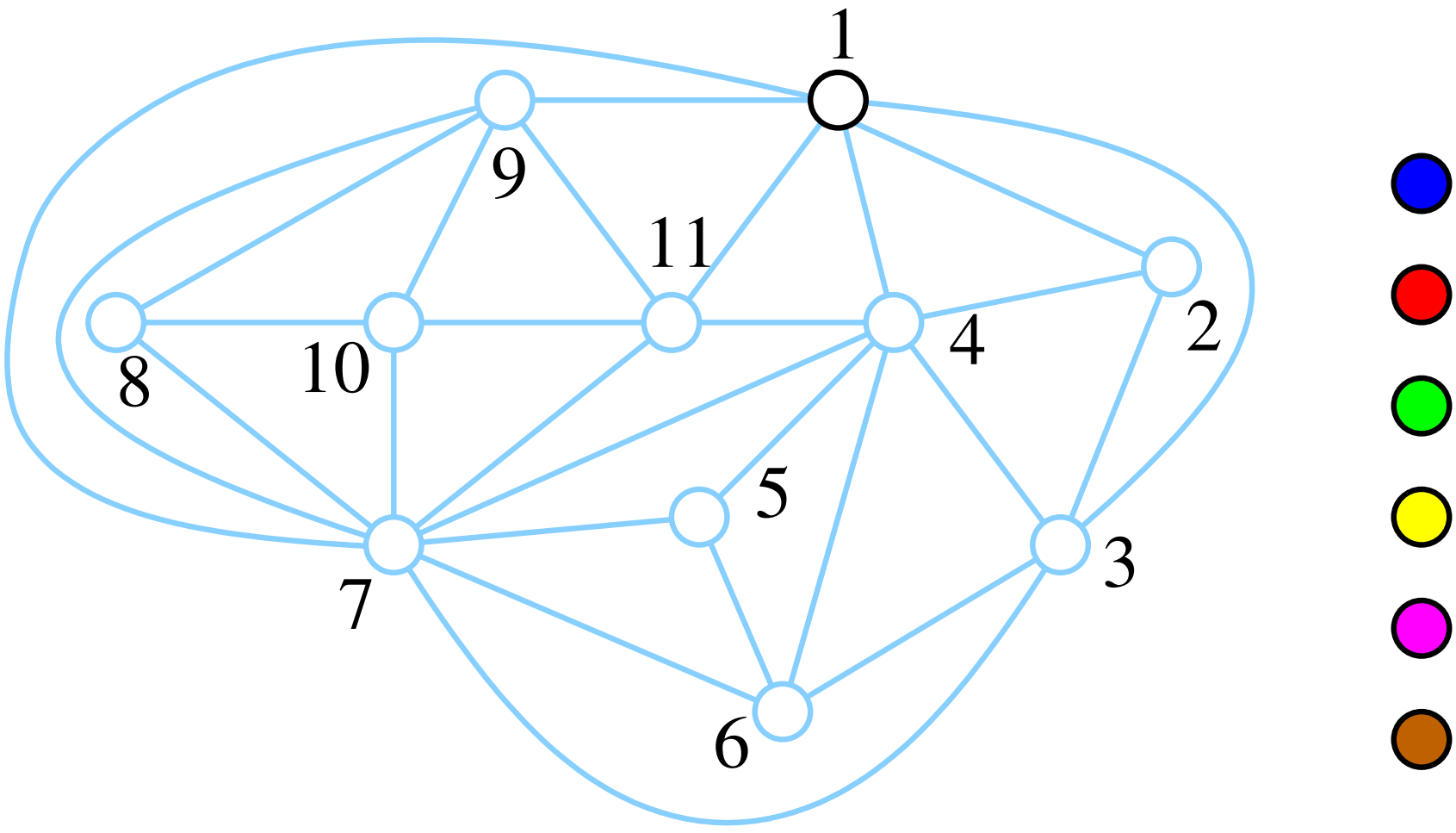
Exemple



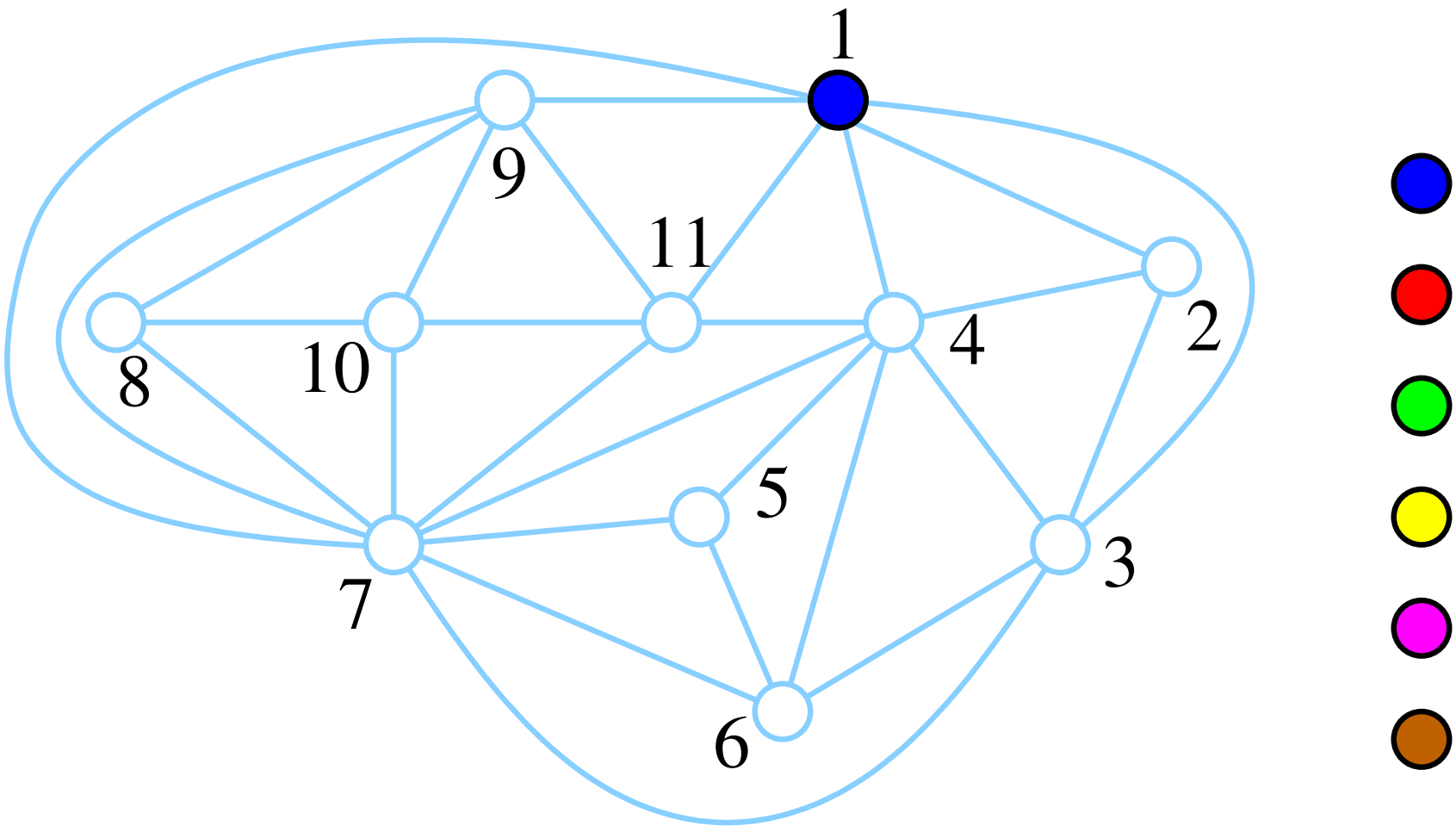
Exemple



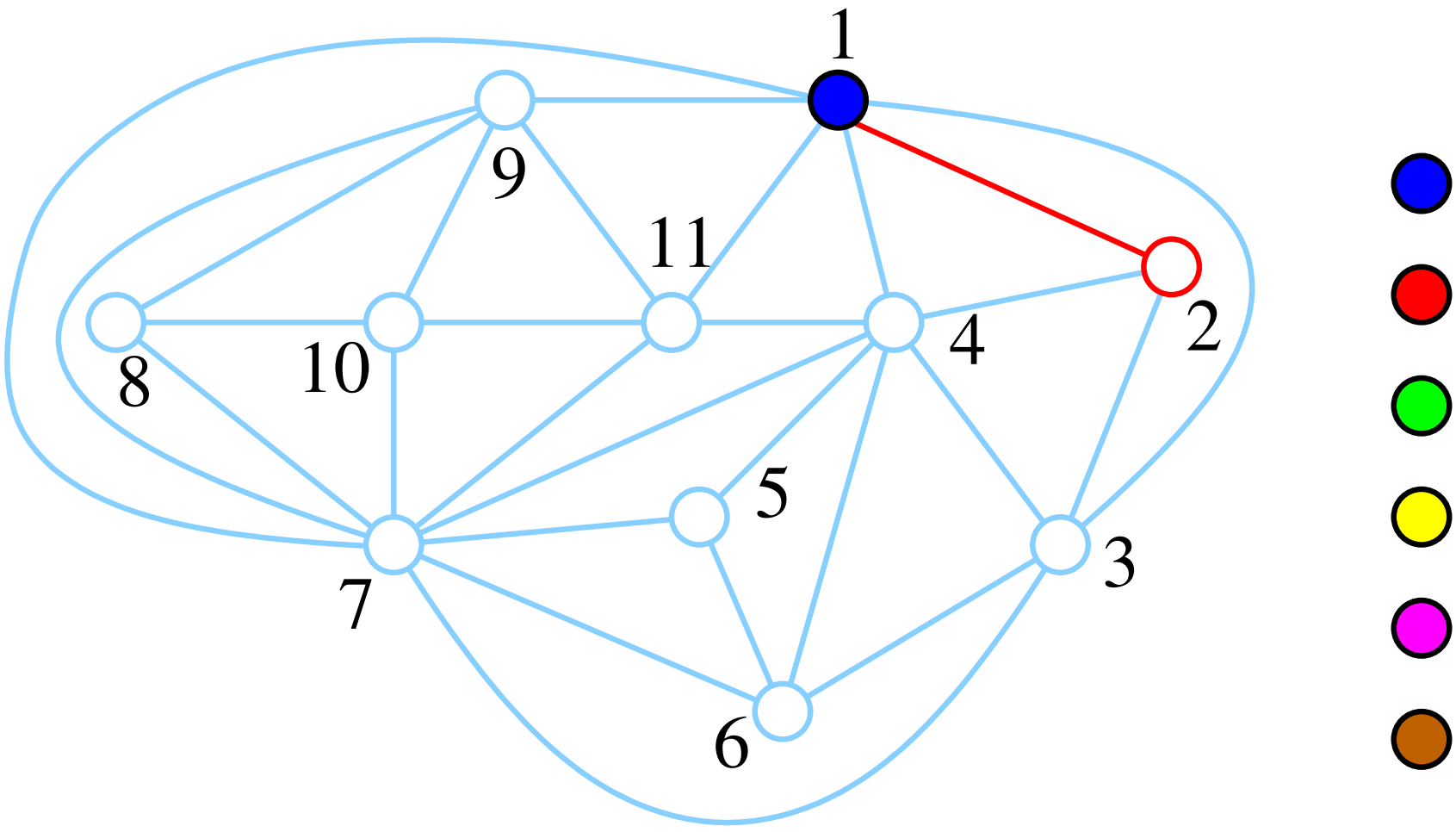
Exemple



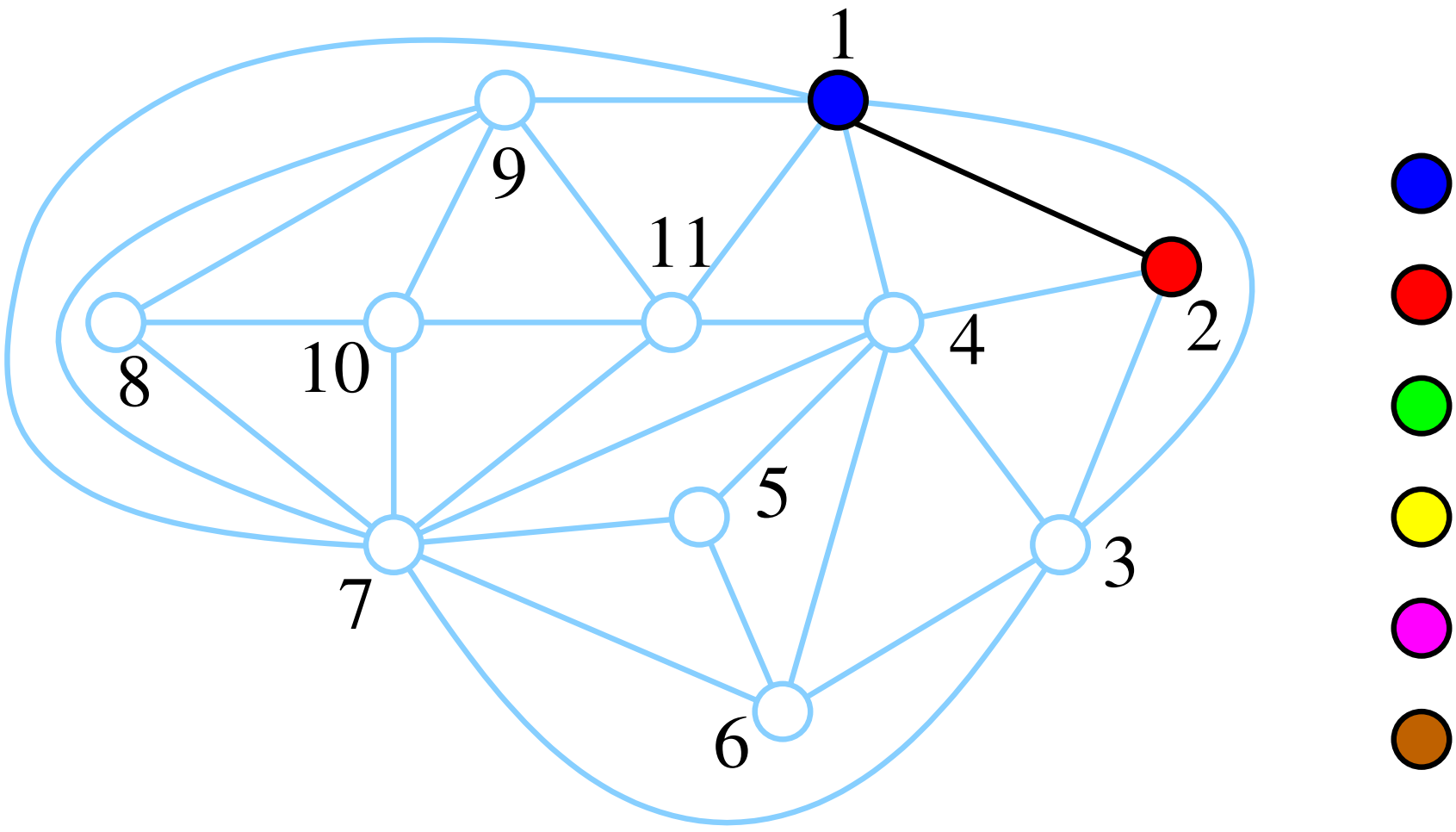
Exemple



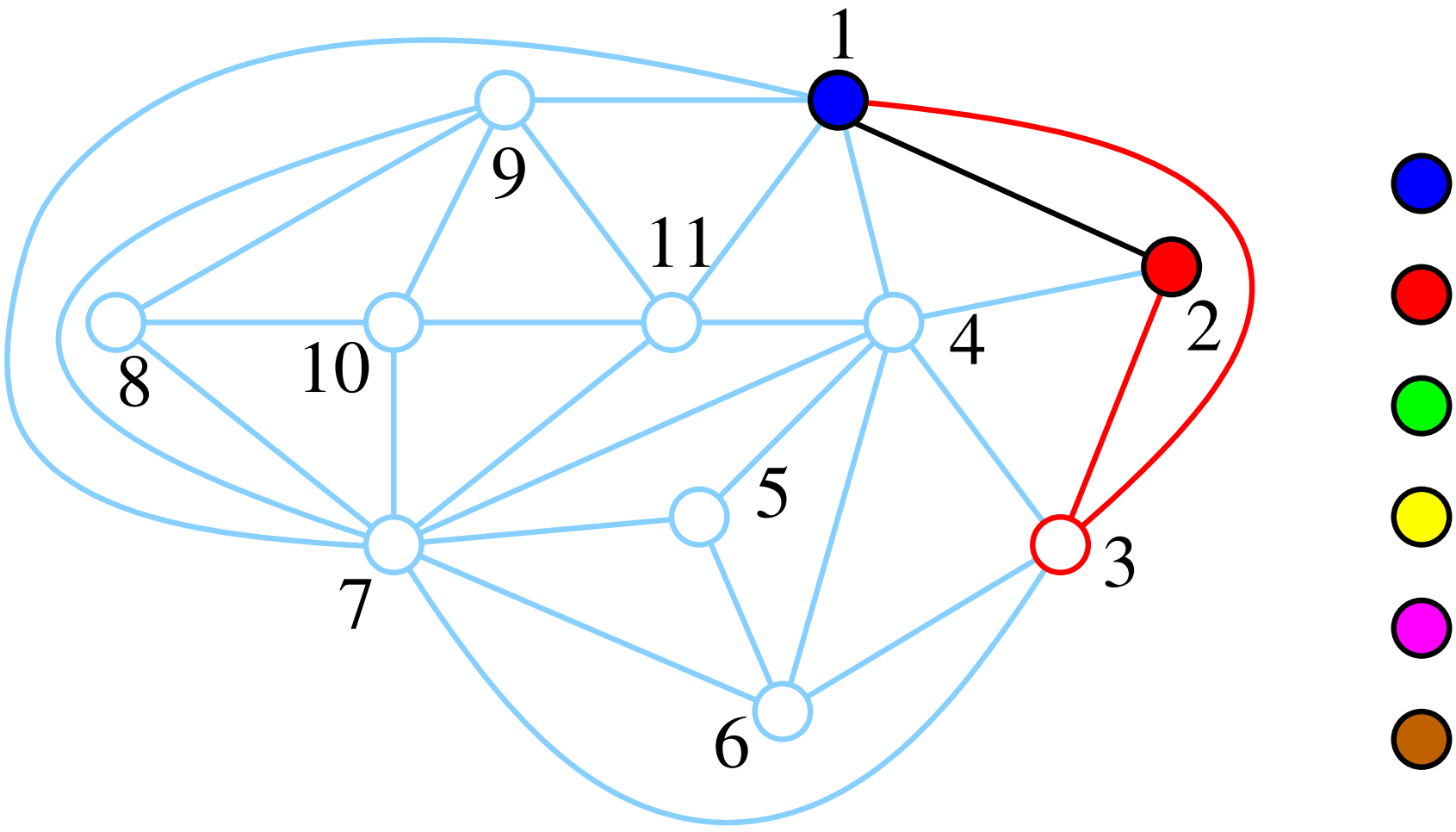
Exemple



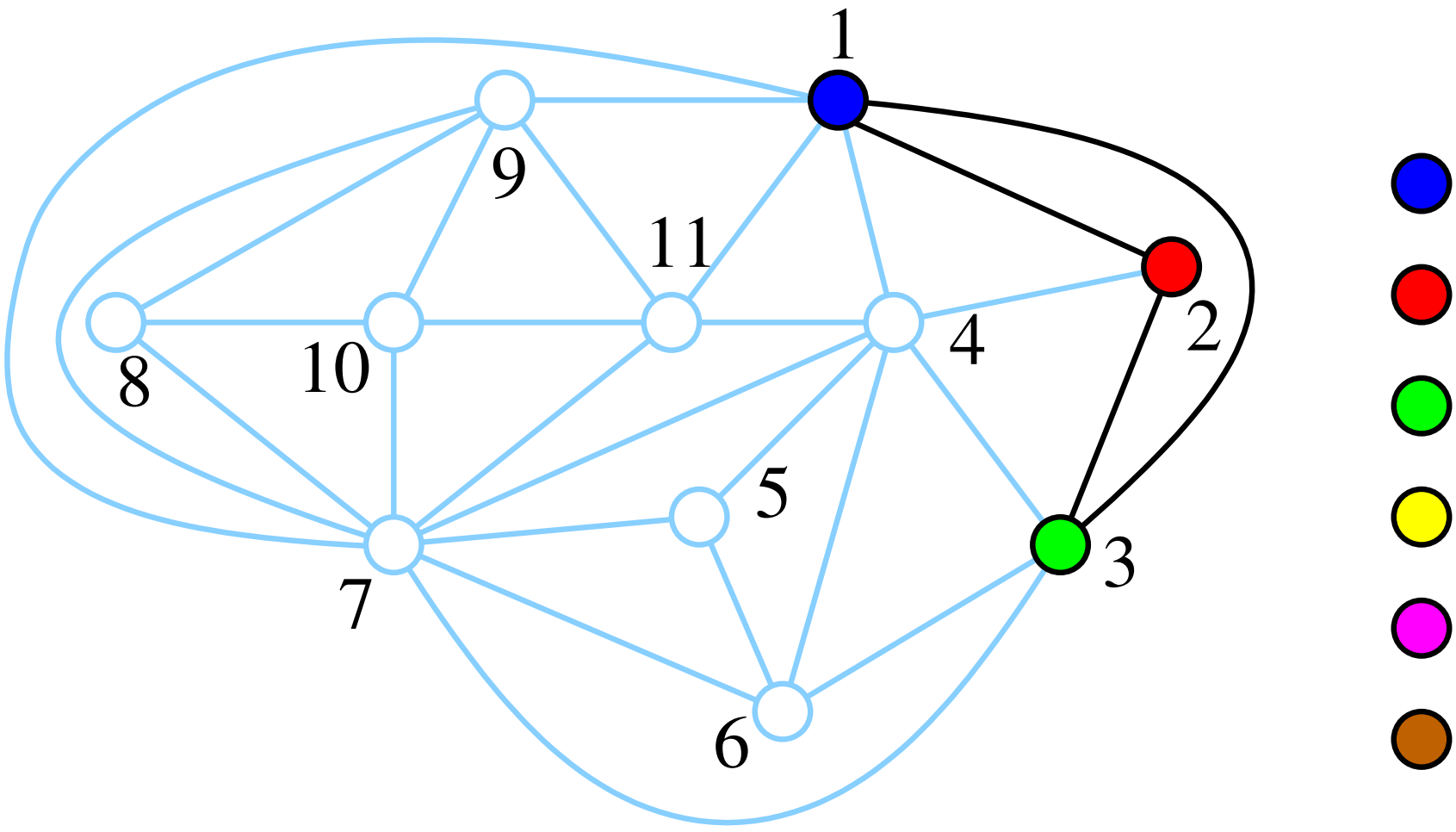
Exemple



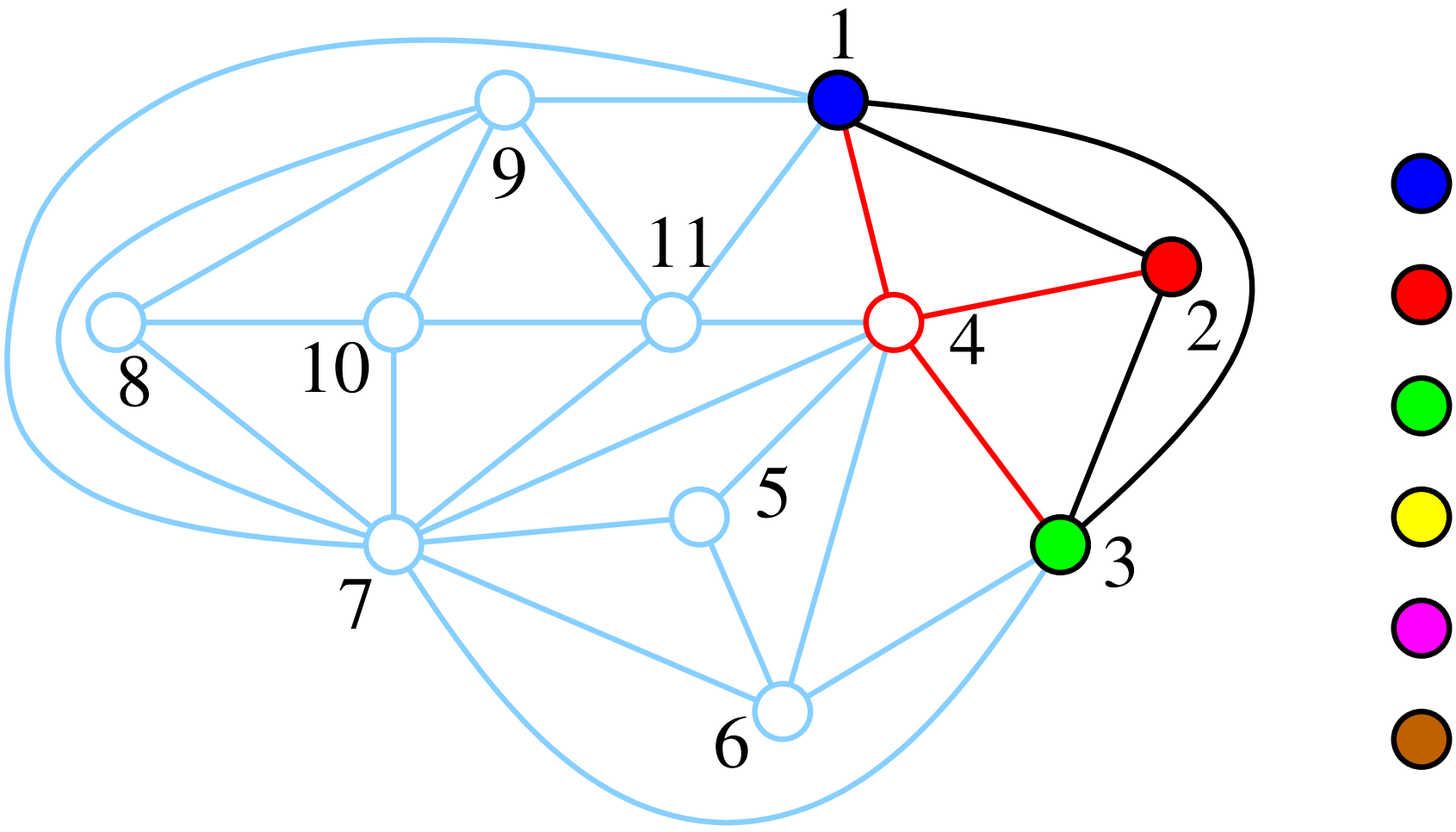
Exemple



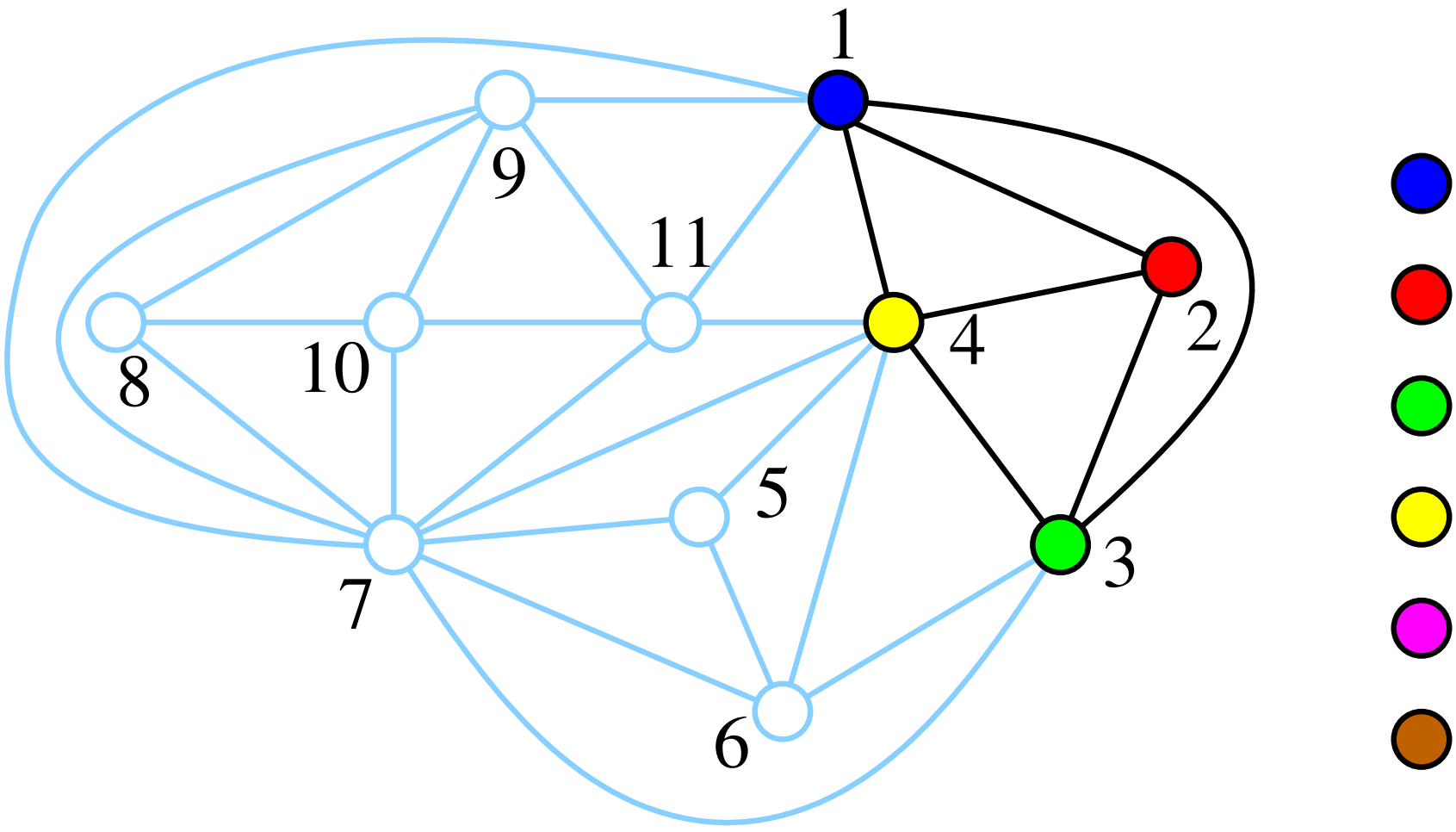
Exemple



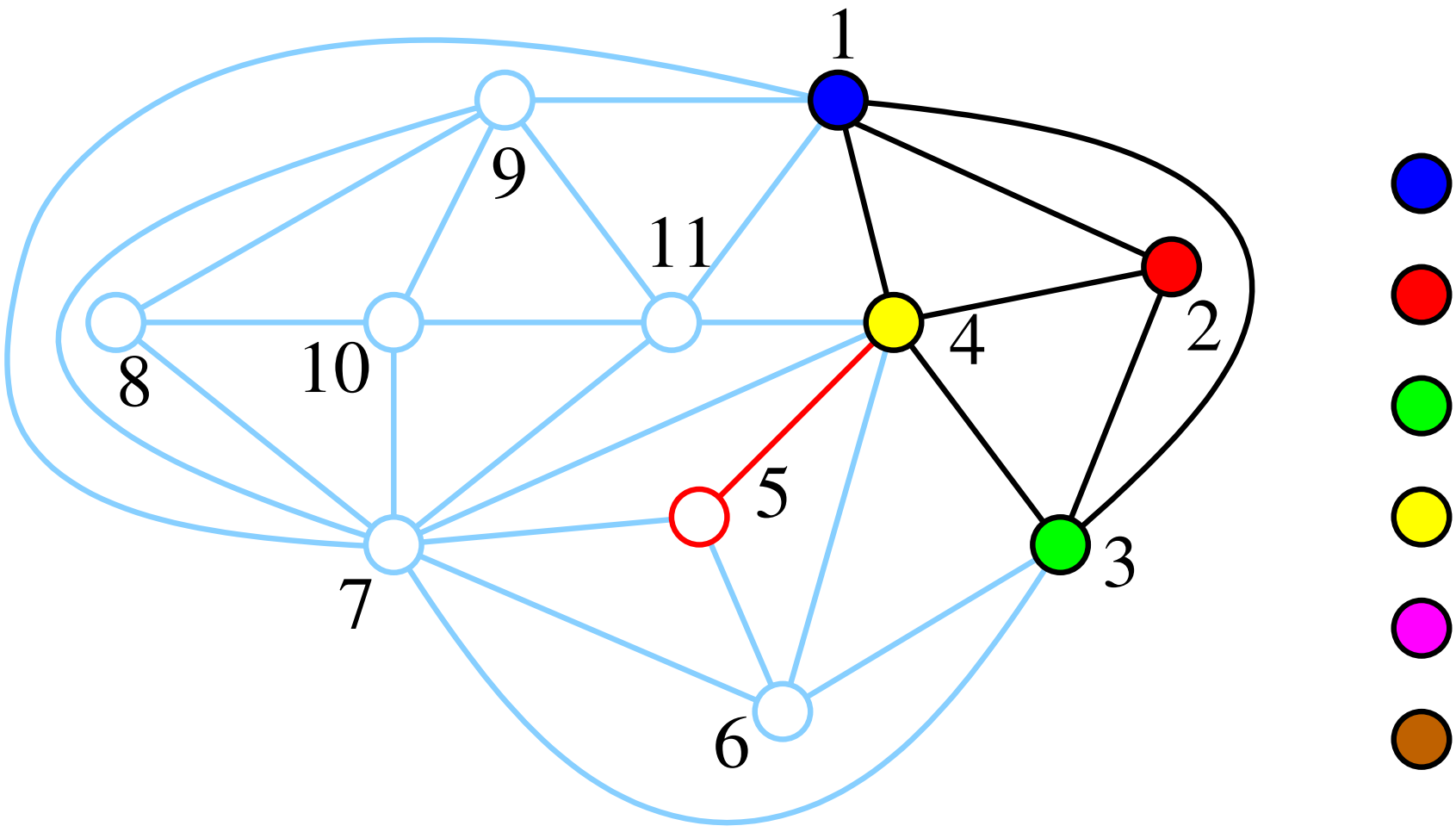
Exemple



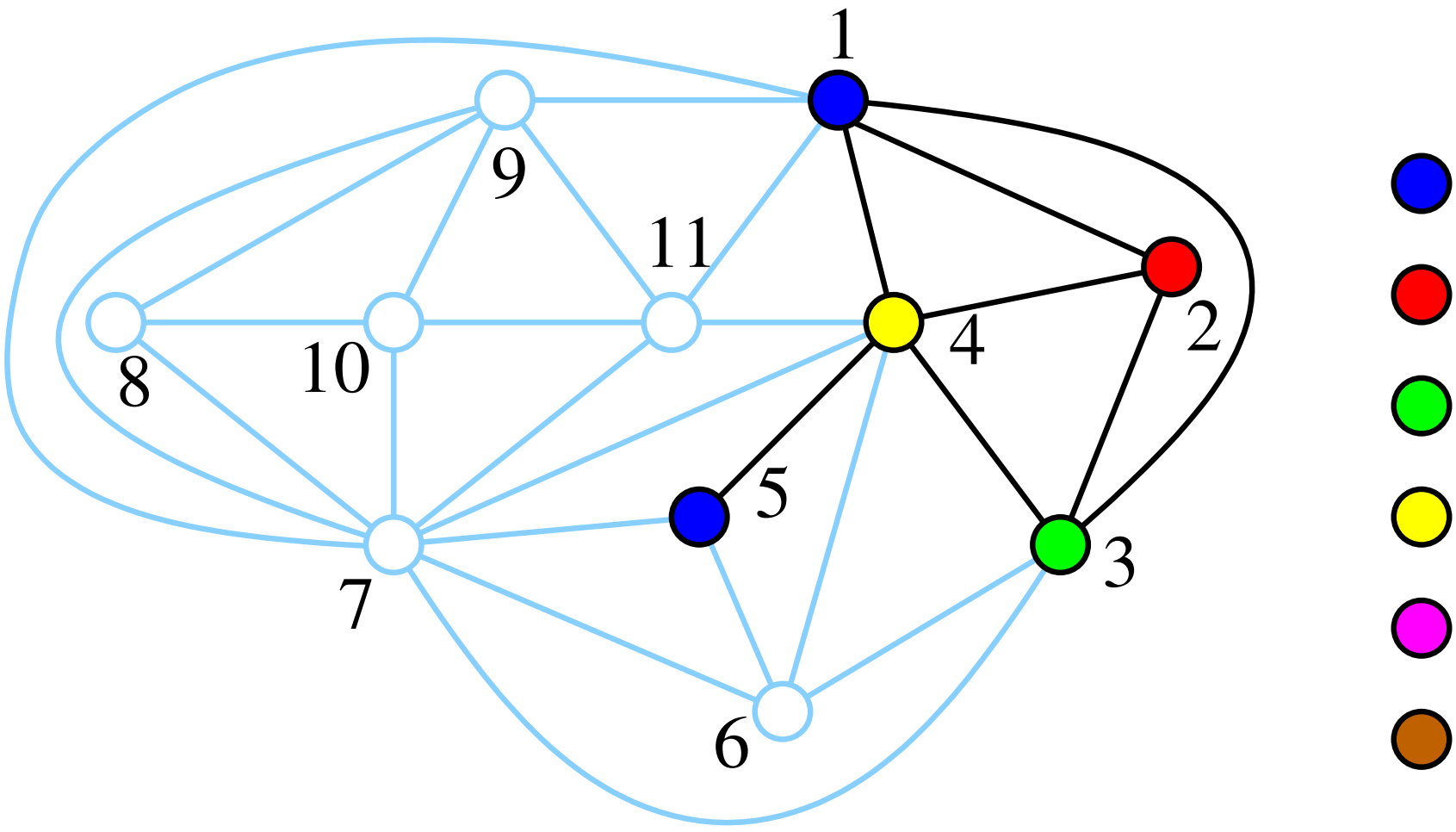
Exemple



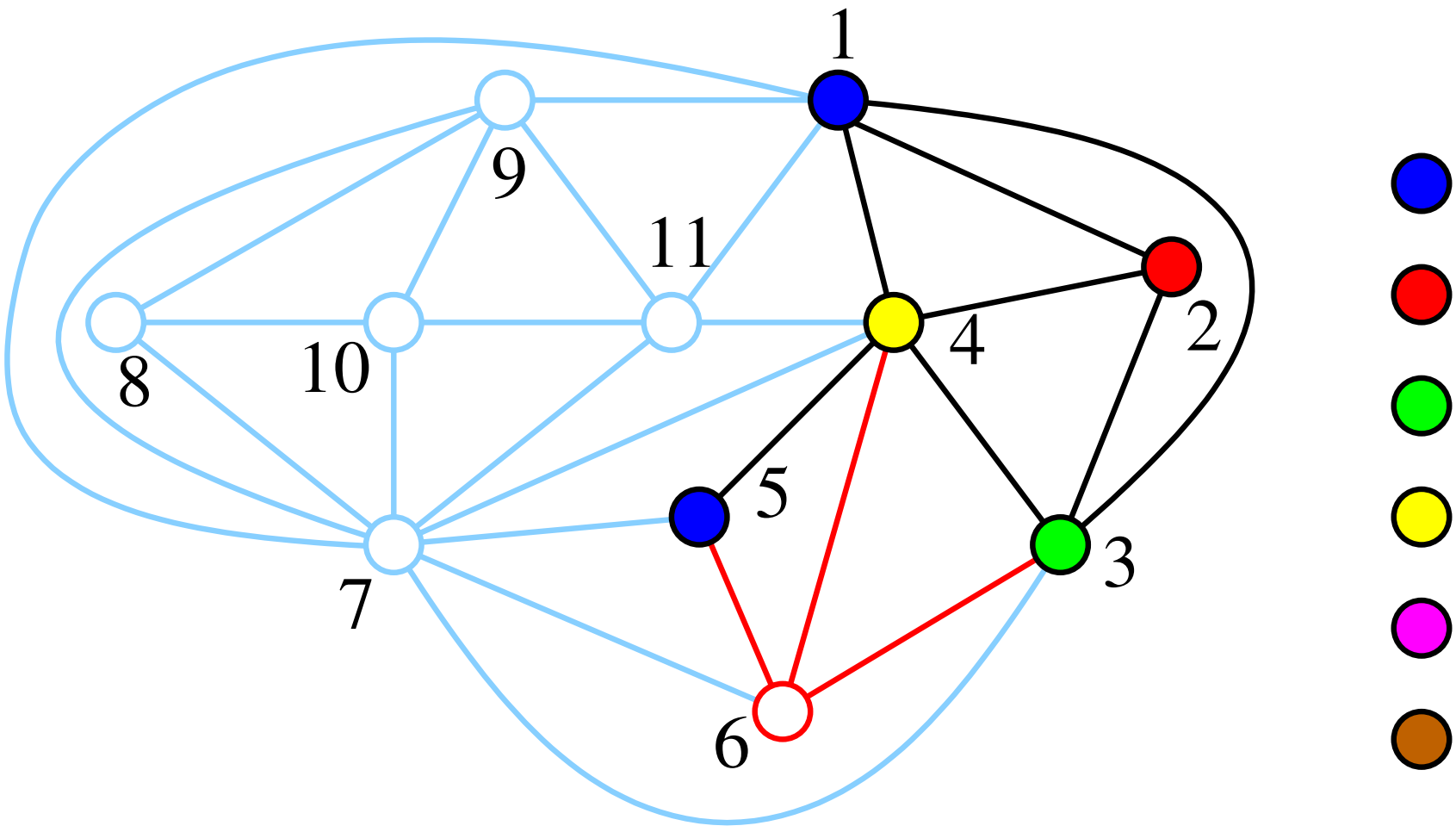
Exemple



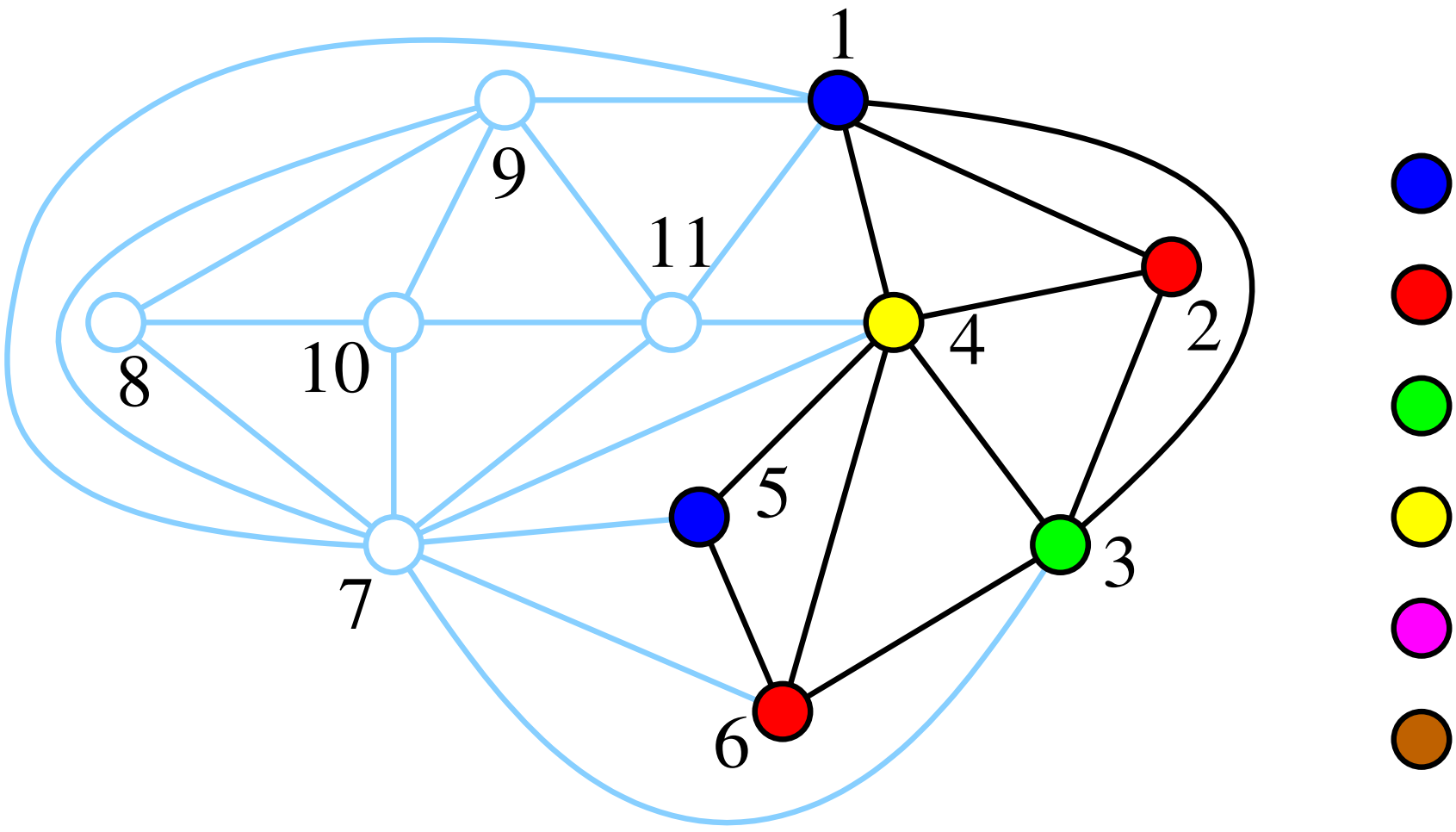
Exemple



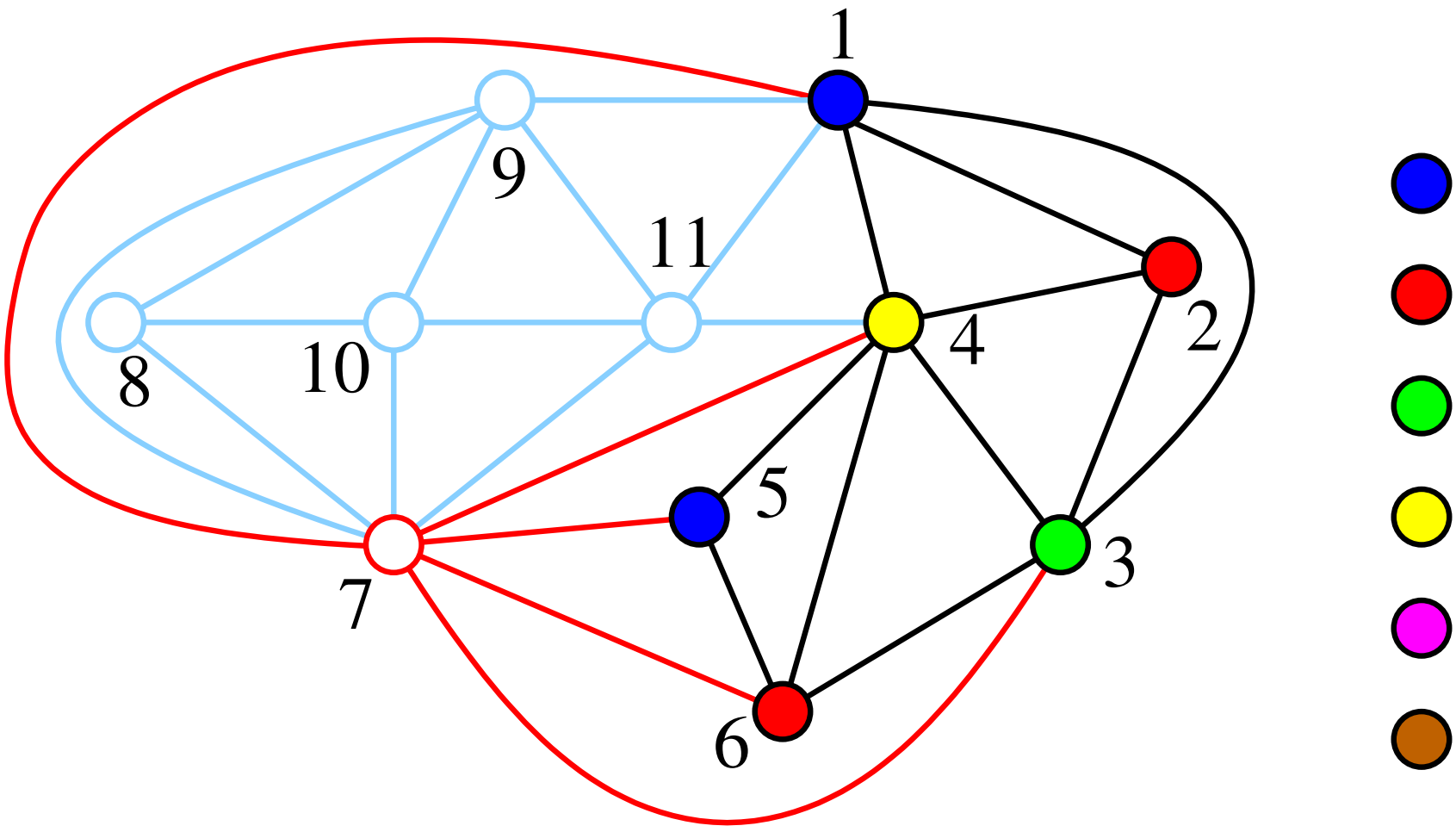
Exemple



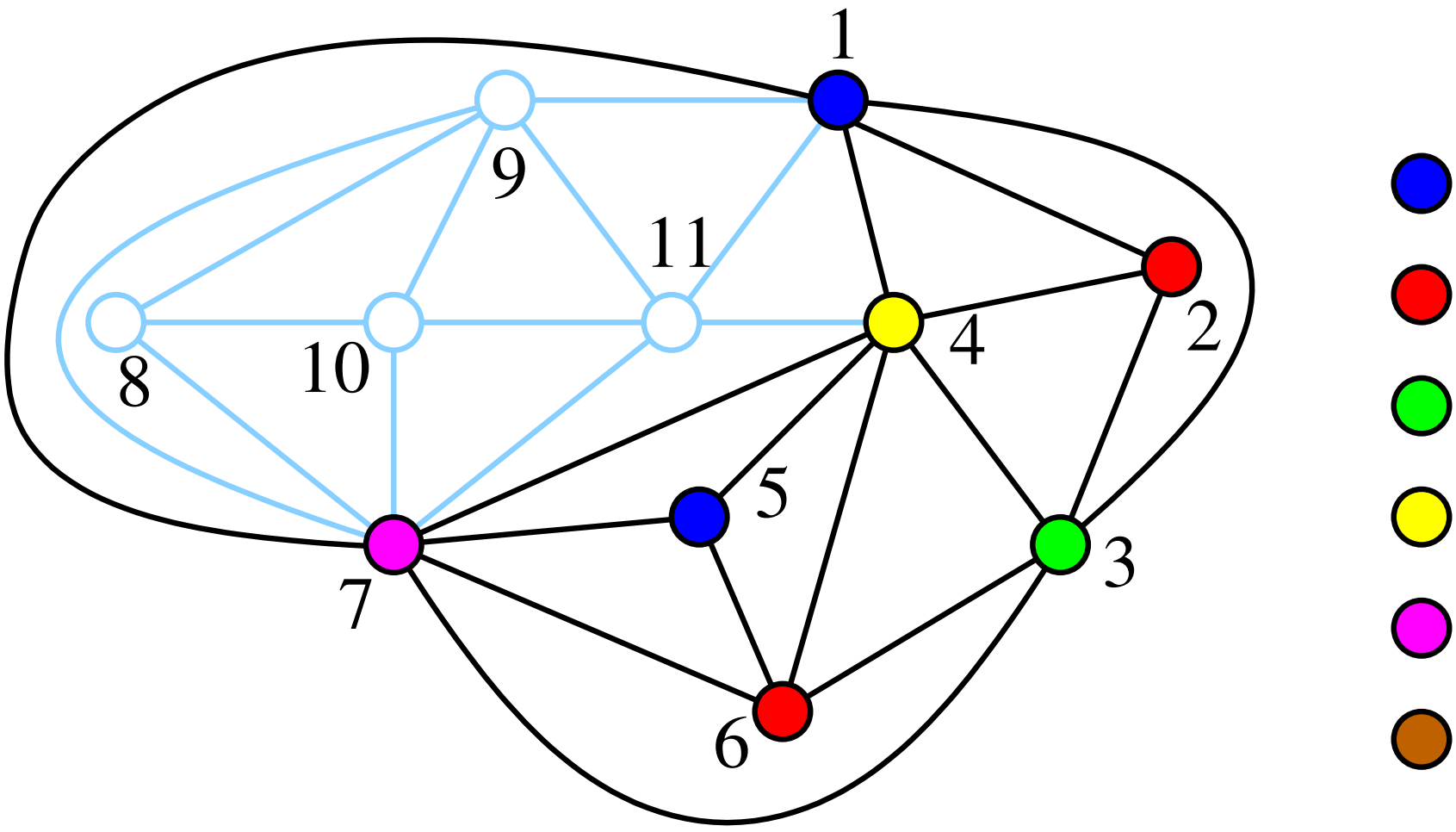
Exemple



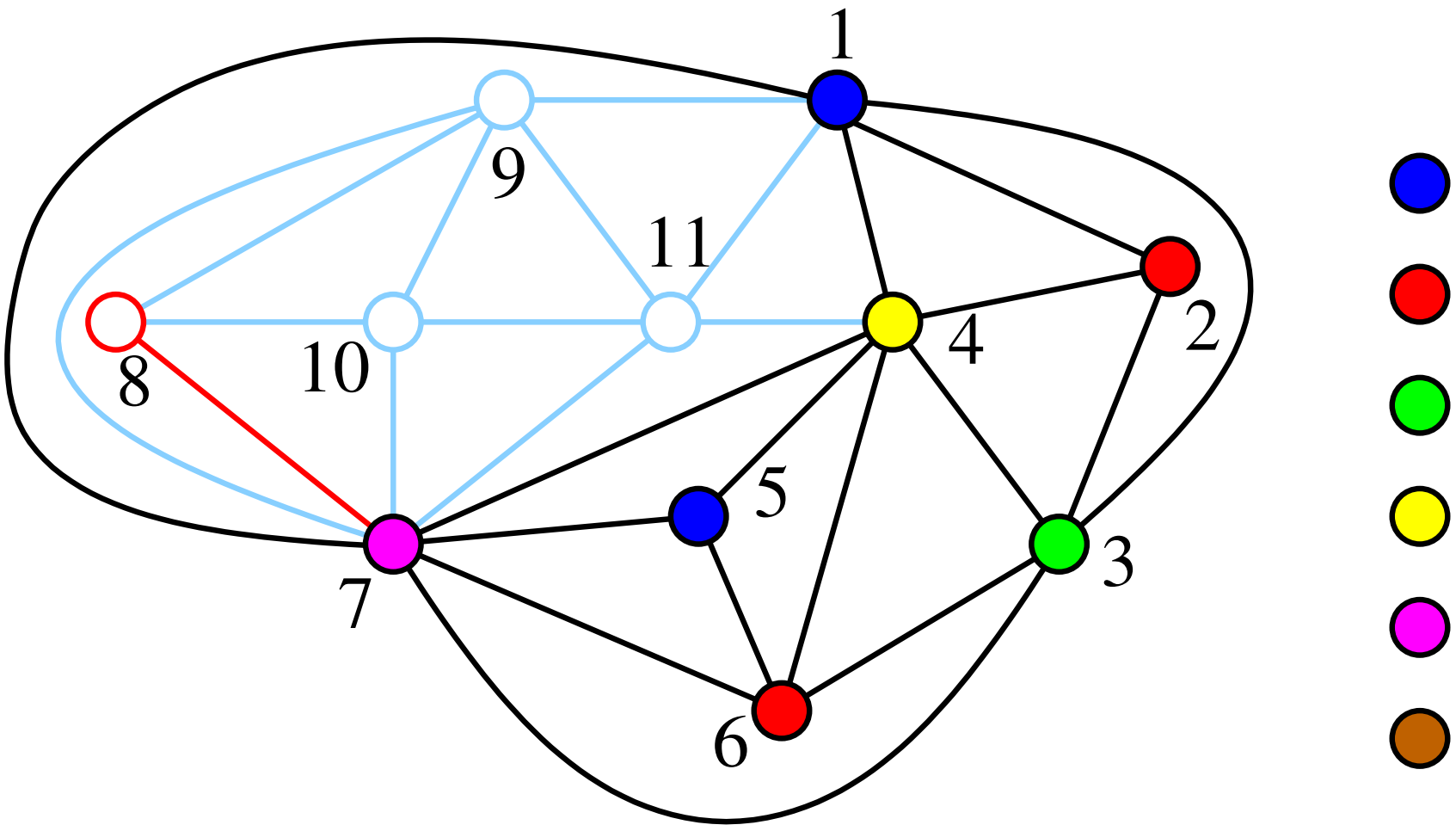
Exemple



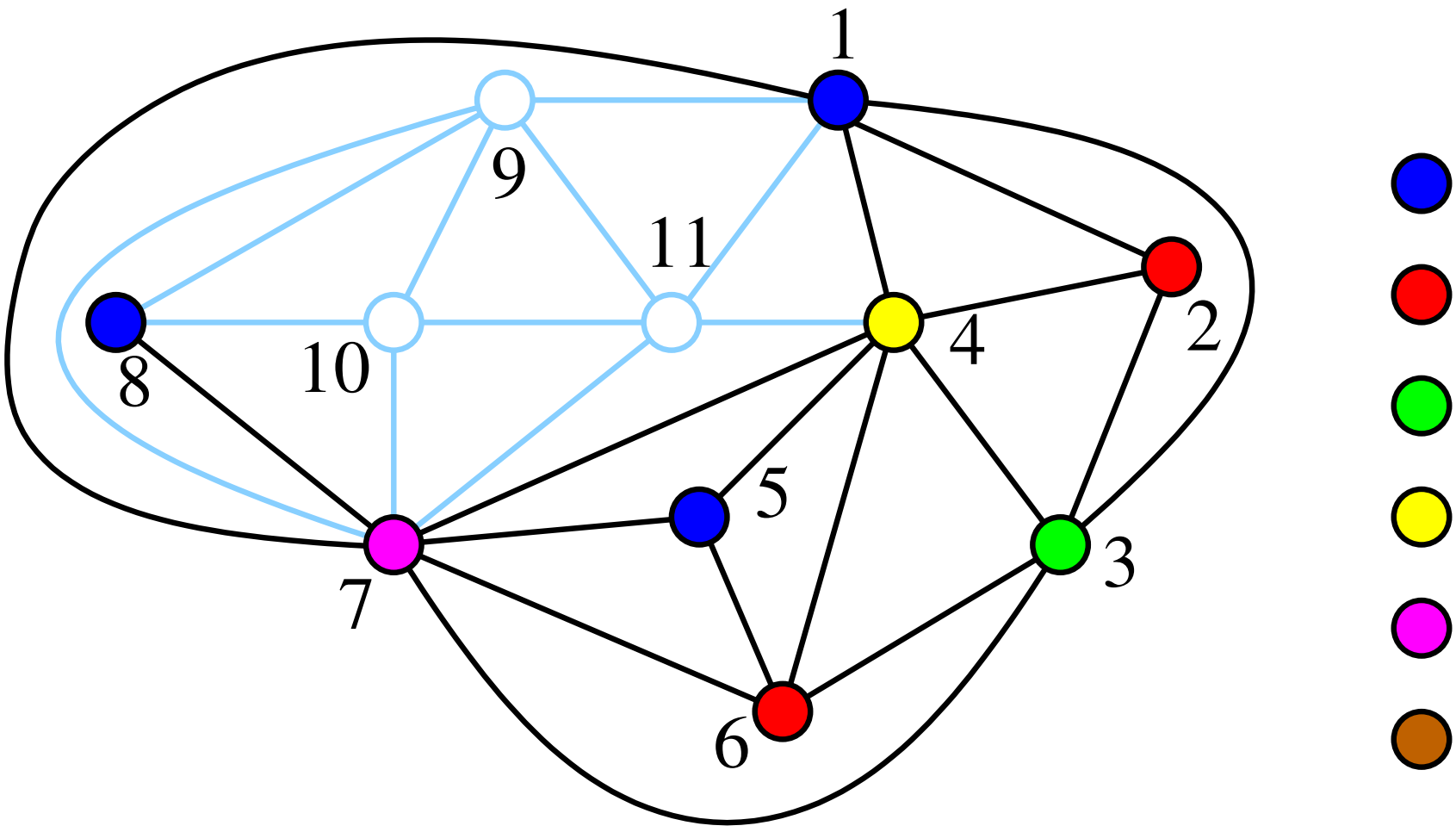
Exemple



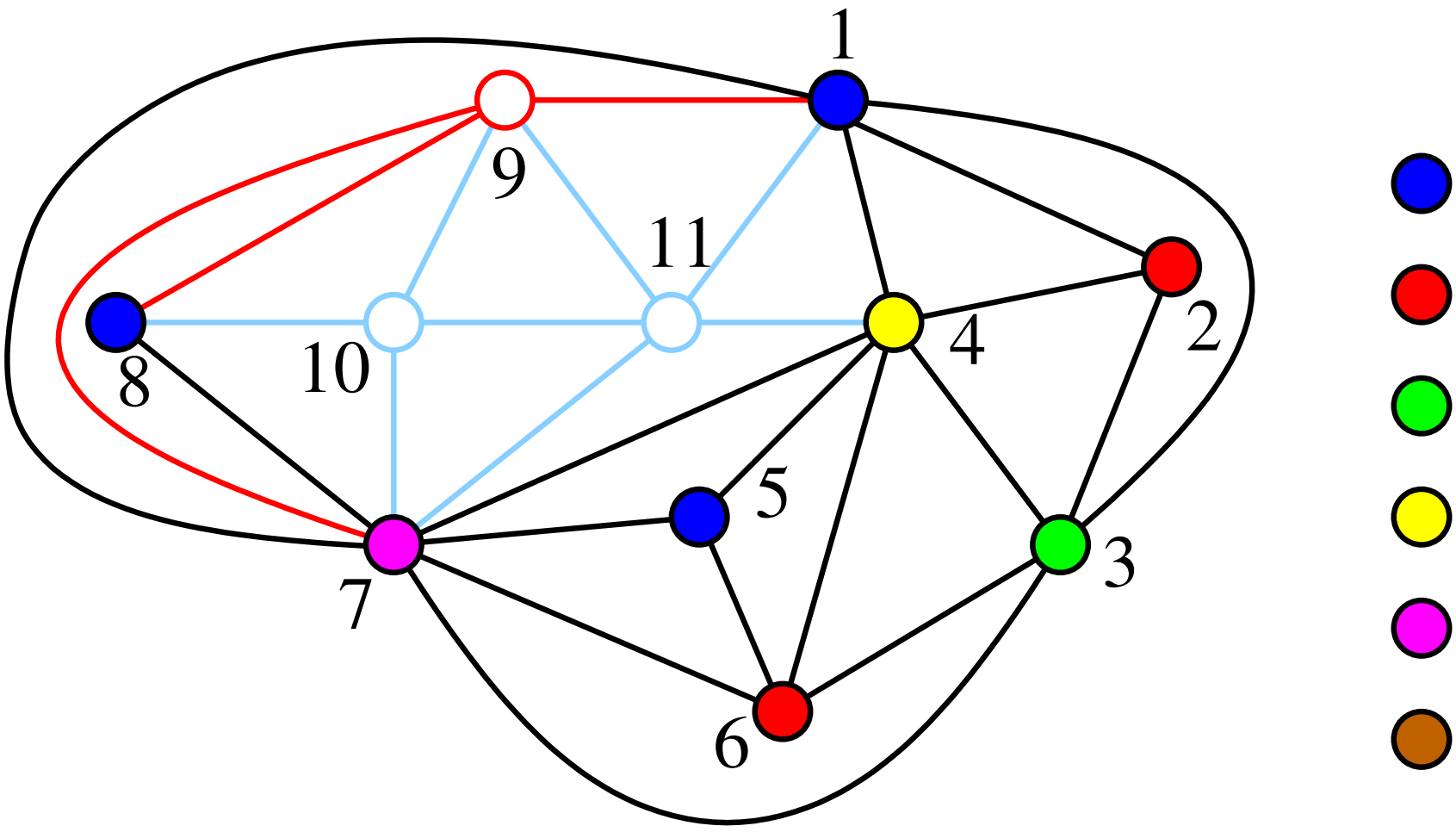
Exemple



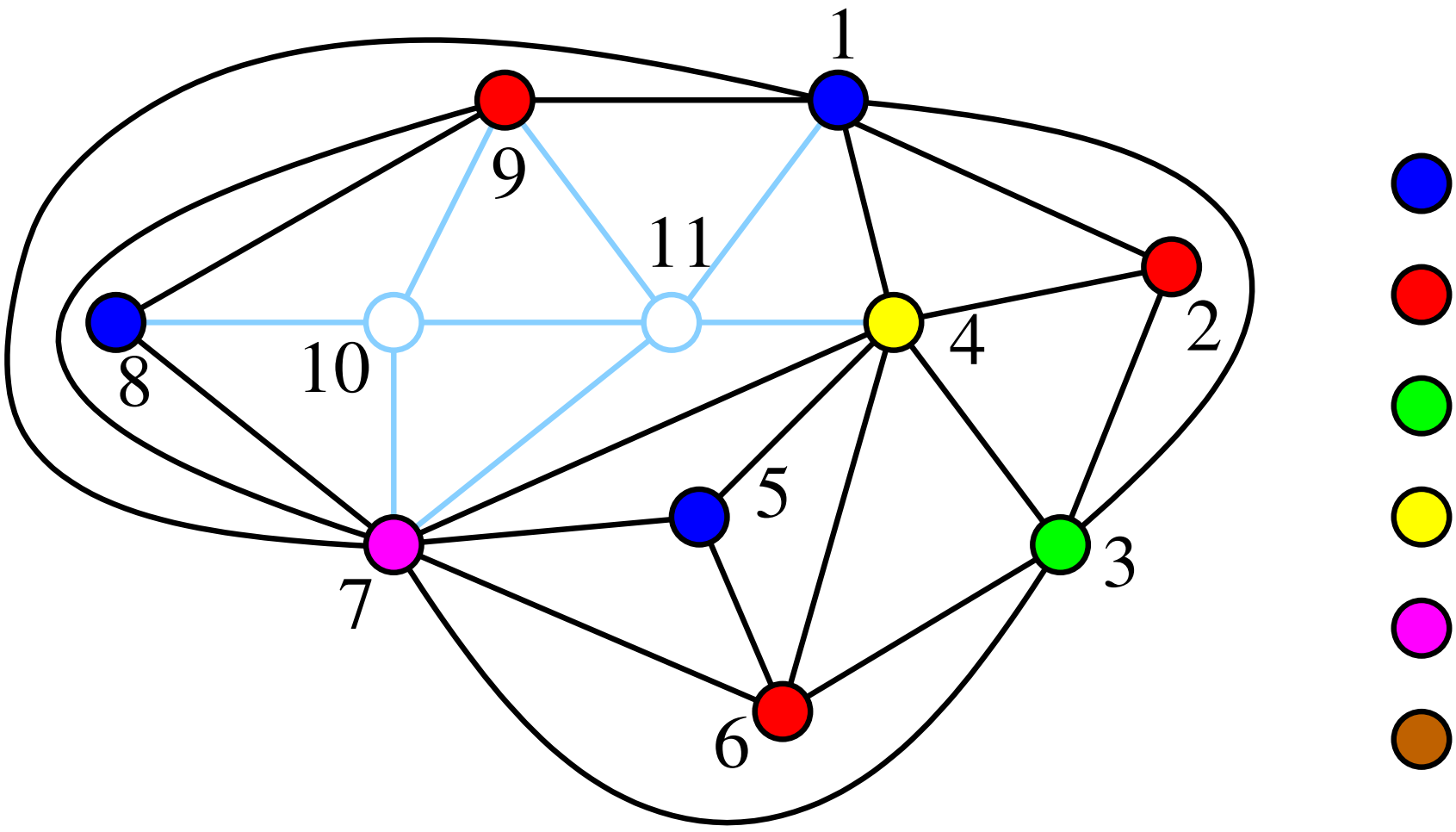
Exemple



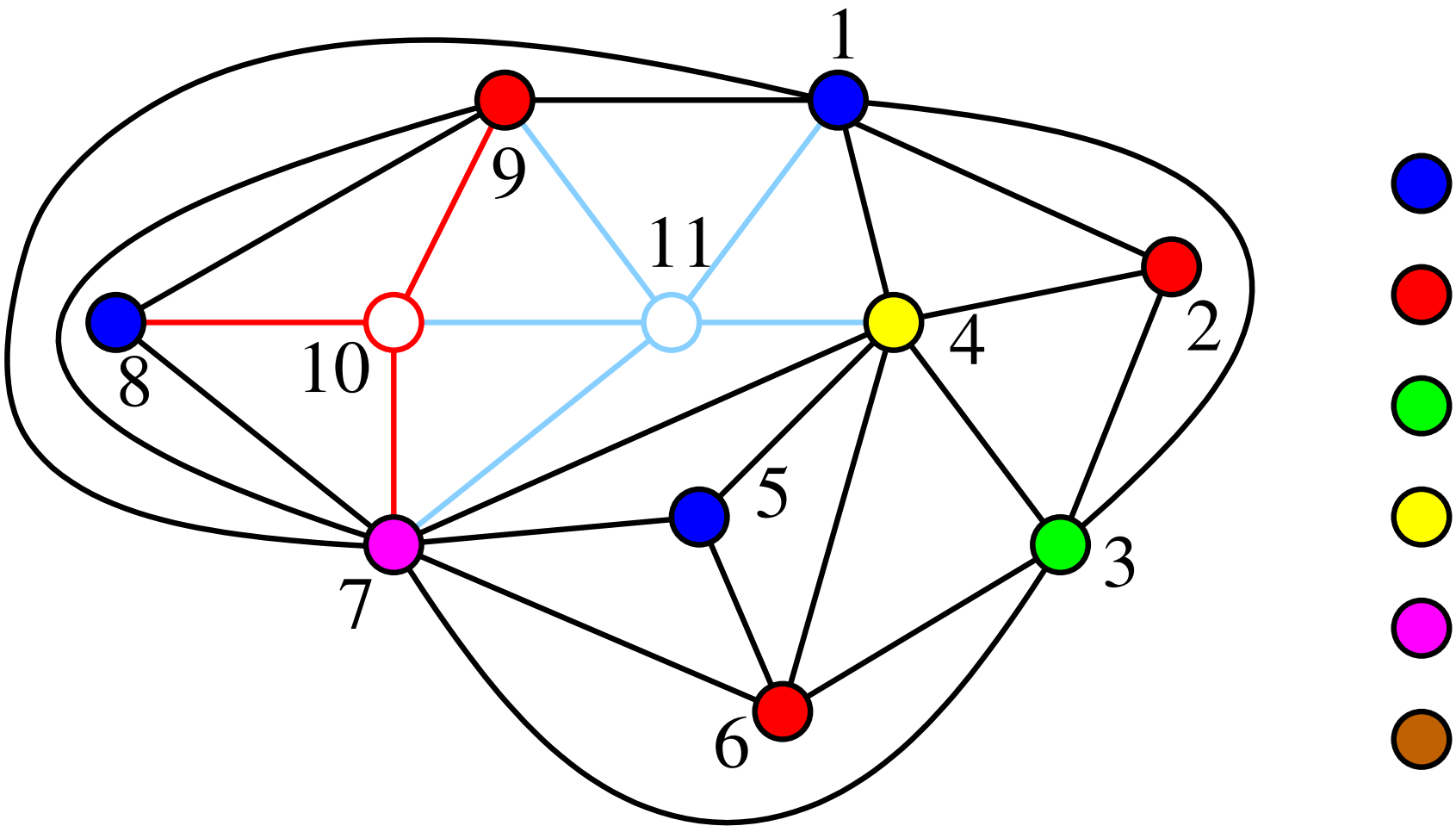
Exemple



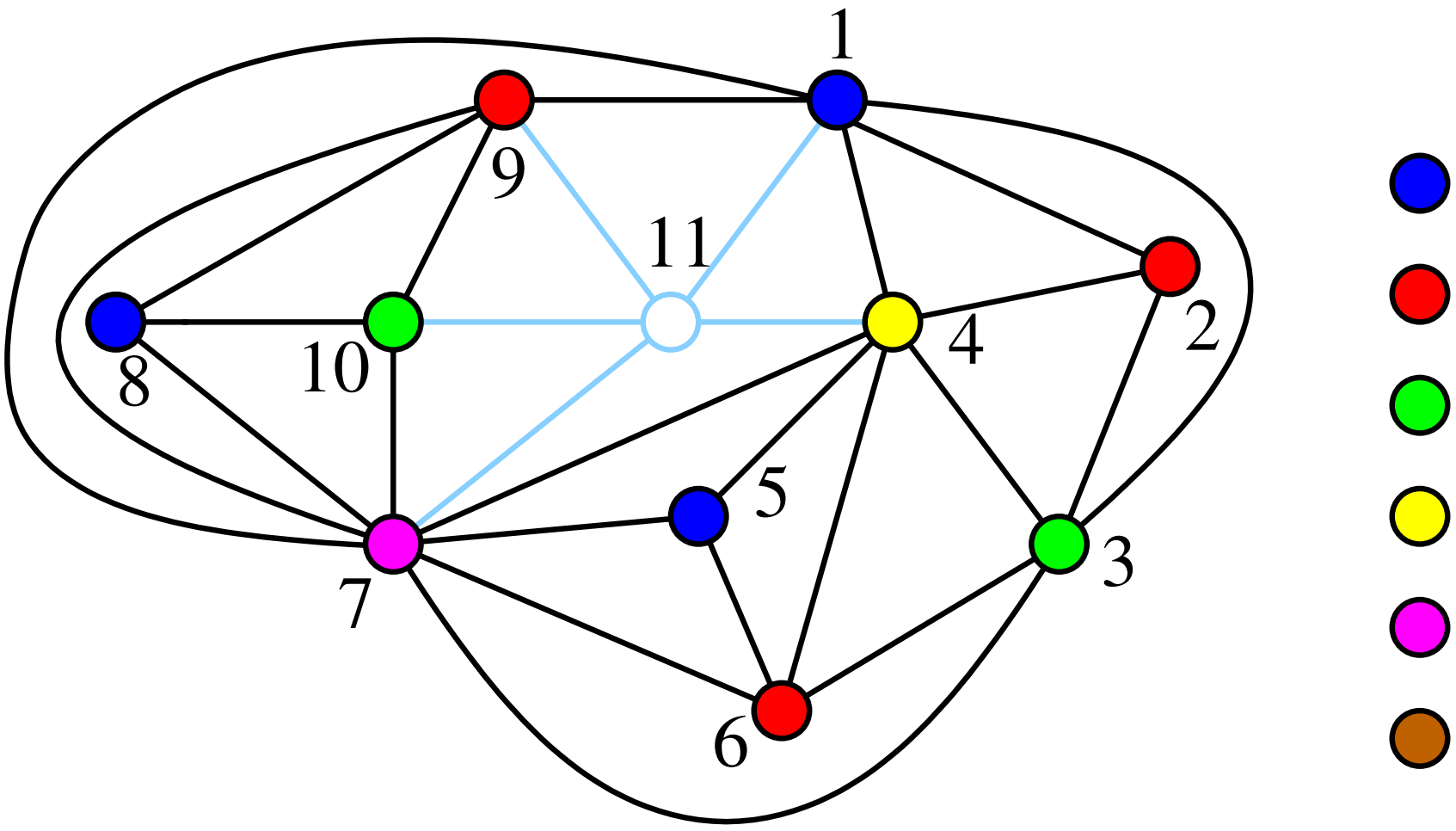
Exemple



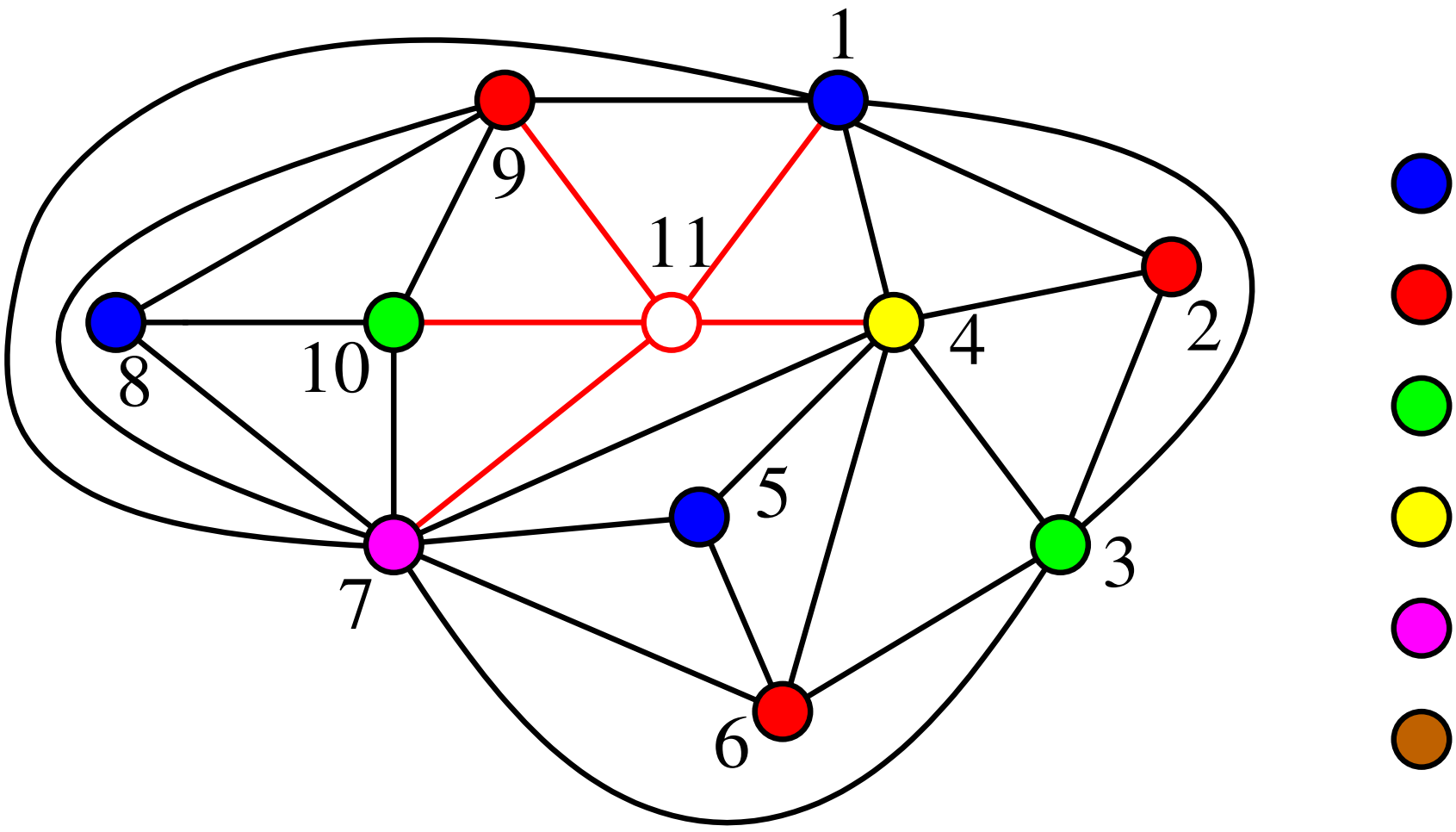
Exemple



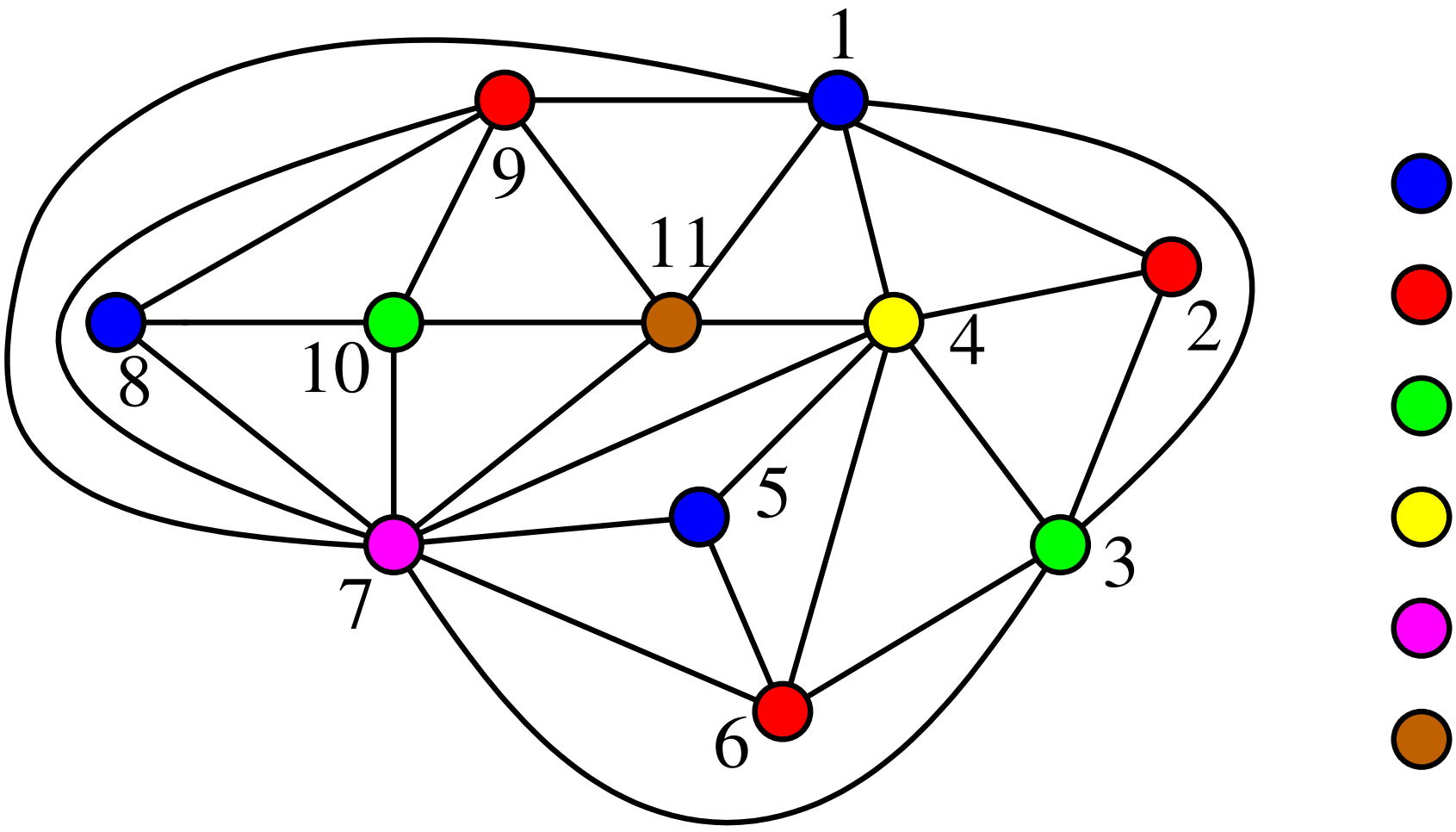
Exemple



Exemple



Exemple



Théorème des 4 couleurs

Kempe en 1879 et **Tait** en 1880 donnent une preuve. Des erreurs sont trouvées par **Heawood** (1890) et **Petersen** (1891).

1976: Preuve par **Appel et Haken**. Réduction à 1478 graphes critiques. Utilisation de l'ordinateur pour résoudre ces cas.
Problème pour la validation:

- Vérification de l'algorithme.
- Vérification de l'implémentation.

1995: Nouvelle preuve par **Robertson, Sanders, Seymour et Thomas**. Réduction à 633 cas.

Le preuve donne un algorithme en $O(n^2)$ pour colorer un graphe planaire.

Colorer un graphe planaire avec 3 couleurs

Quel est le meilleur temps d'exécution (au pire) possible pour un algorithme résolvant le problème suivant:

Entrée: un graphe G (planaire).
Question: peut-on colorer G avec 3 couleurs?

Il y a $3^n = \overbrace{3 \times 3 \times \cdots \times 3}^{n \text{ fois}}$ colorations (non forcément propres) donc on peut le faire en $O^*(3^n)$.

Conjecture: C'est impossible en temps polynomial.

C'est $P \neq NP$, une des conjectures du millénaire.
1 million de dollars est offert pour qui la montre ou la réfute.

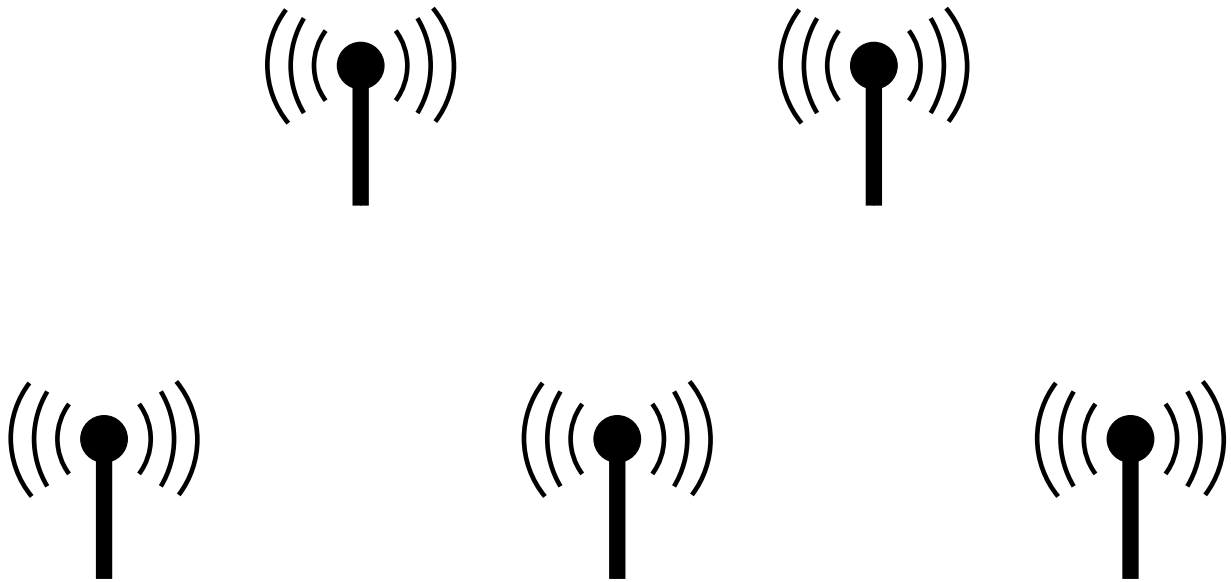
Le même problème avec 2 couleurs se fait facilement en temps polynomial.

Applications de la coloration

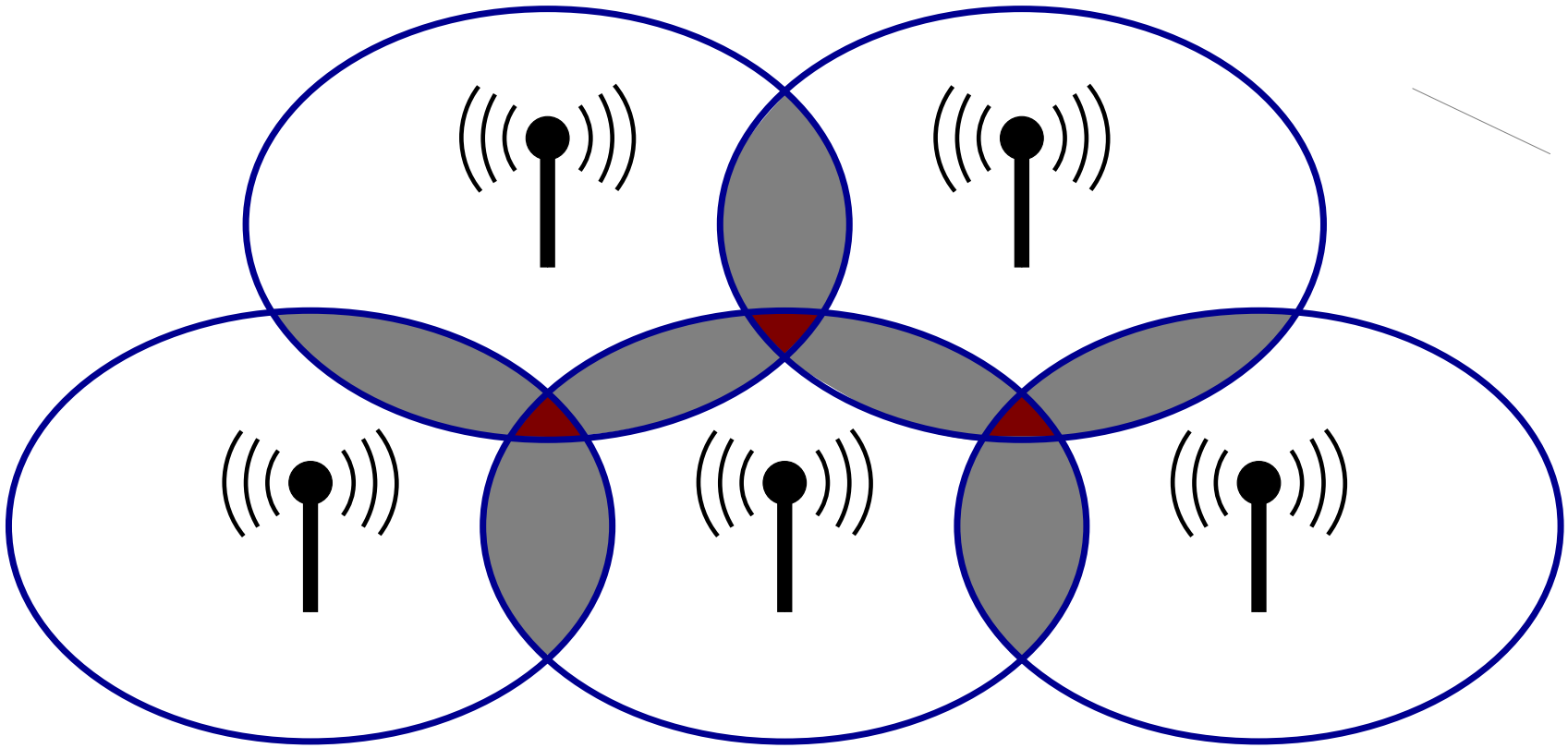
- **Ordonnancement.**
  Sommets = tâches.
  Arêtes entre des tâches concurrentes.
  Couleurs = intervalles de temps.
  Coloration du graphe = ordonnancement possible des tâches.
  Minimiser le nombre de couleurs = minimiser le temps pour effectuer toutes les tâches.

- **Allocation de fréquences.**
  Sommets = transmetteurs.
  Arêtes entre des transmetteurs dont les zones s'intersectent.
  Couleurs = fréquences.
  Coloration du graphe = allocation de fréquences sans interférences.
  Minimiser le nombre de couleurs = minimiser la largeur de bande nécessaire.

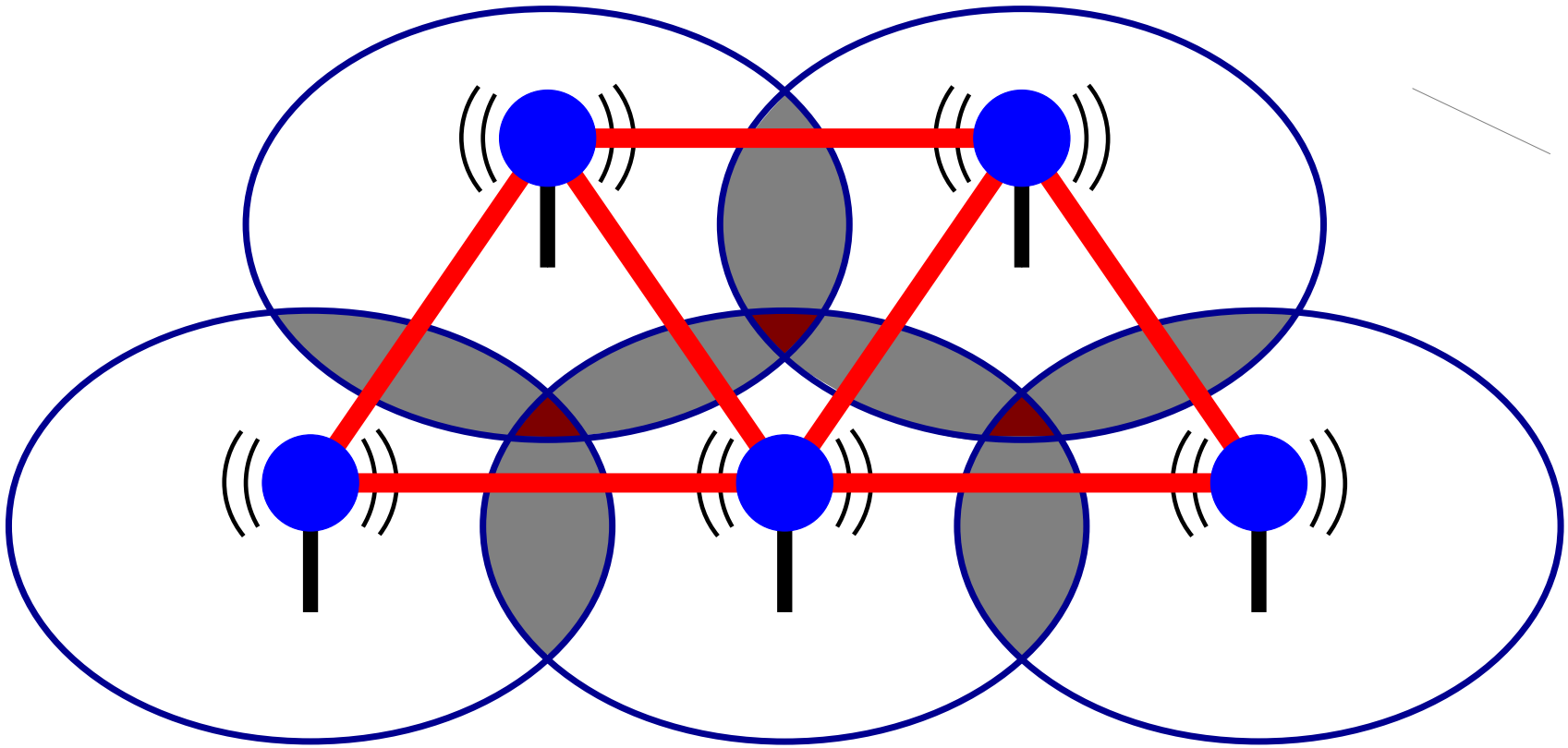
Allocation de fréquences



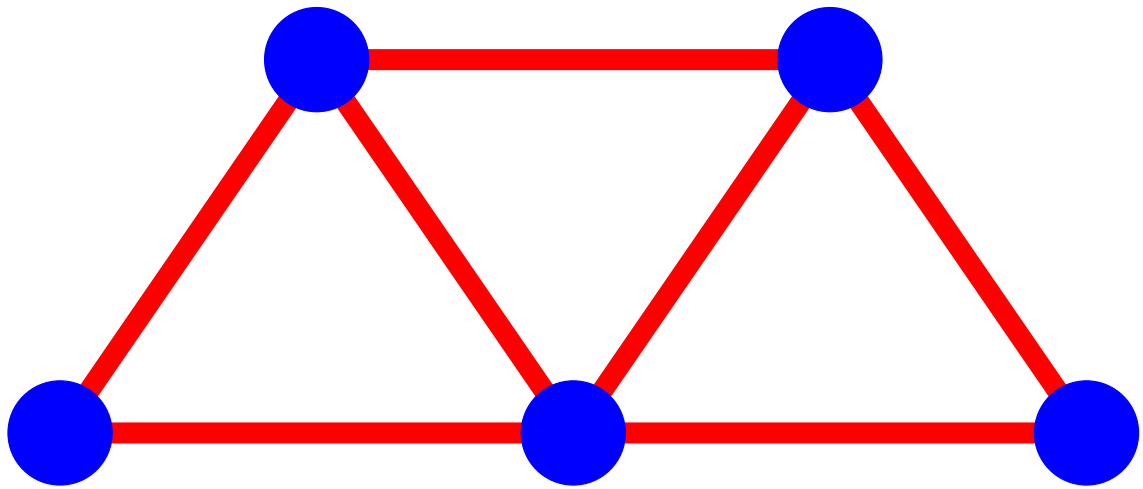
Allocation de fréquences



Allocation de fréquences



Allocation de fréquences



BONUS: Tour de cartes

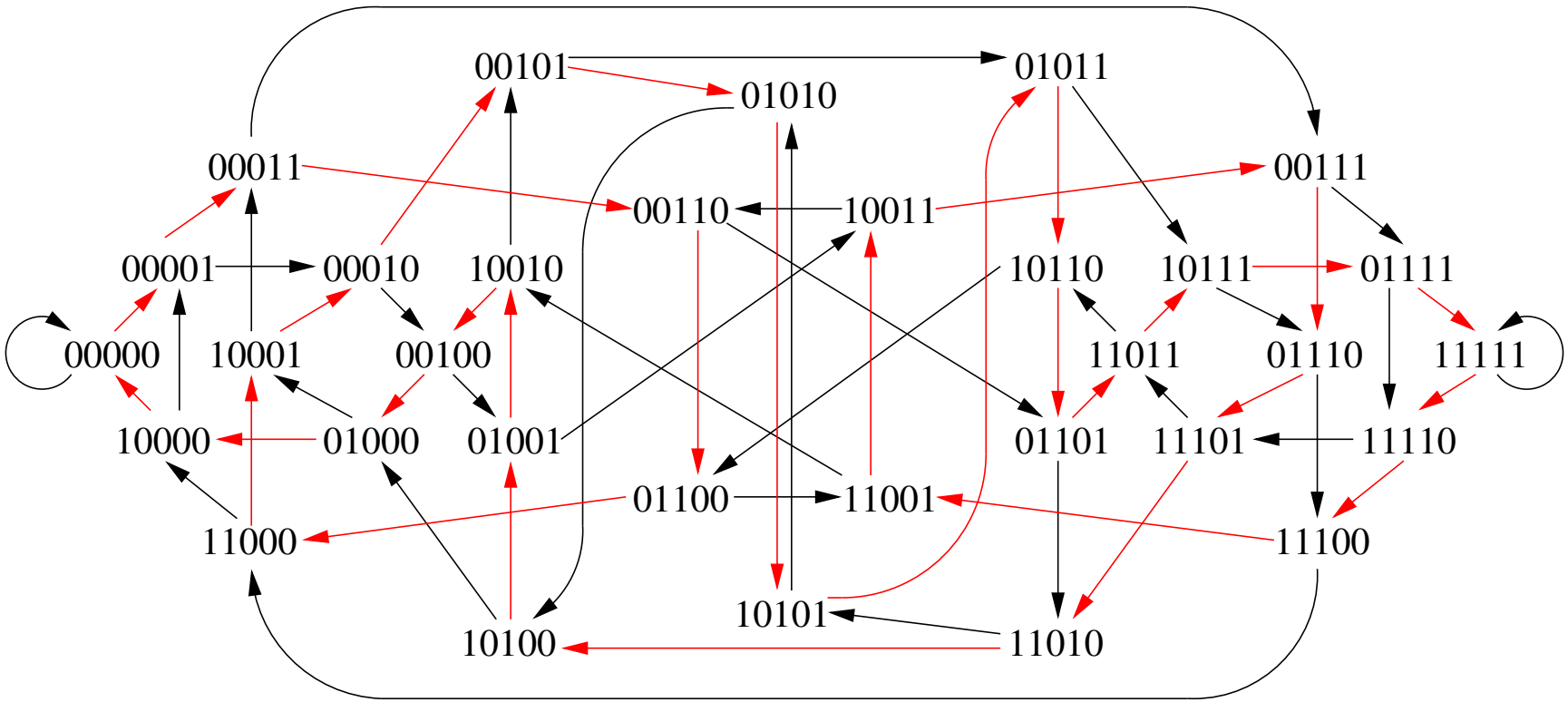
Tour de cartes – le truc

A chaque carte, on associe un nombre en binaire à 5 bits comme suit:

- Les deux derniers bits représentent la couleur de la carte. 00=coeur, 10=carreau, 01 =trèfle et 11=pique. Ainsi le tout dernier bit nous dit si la carte est rouge ou noire.
- Les trois premiers bits représentent la valeur de la carte. 000=as; 001=7; 010=8; 011=9; 100=10; 101=valet; 110 =dame; 111=roi.

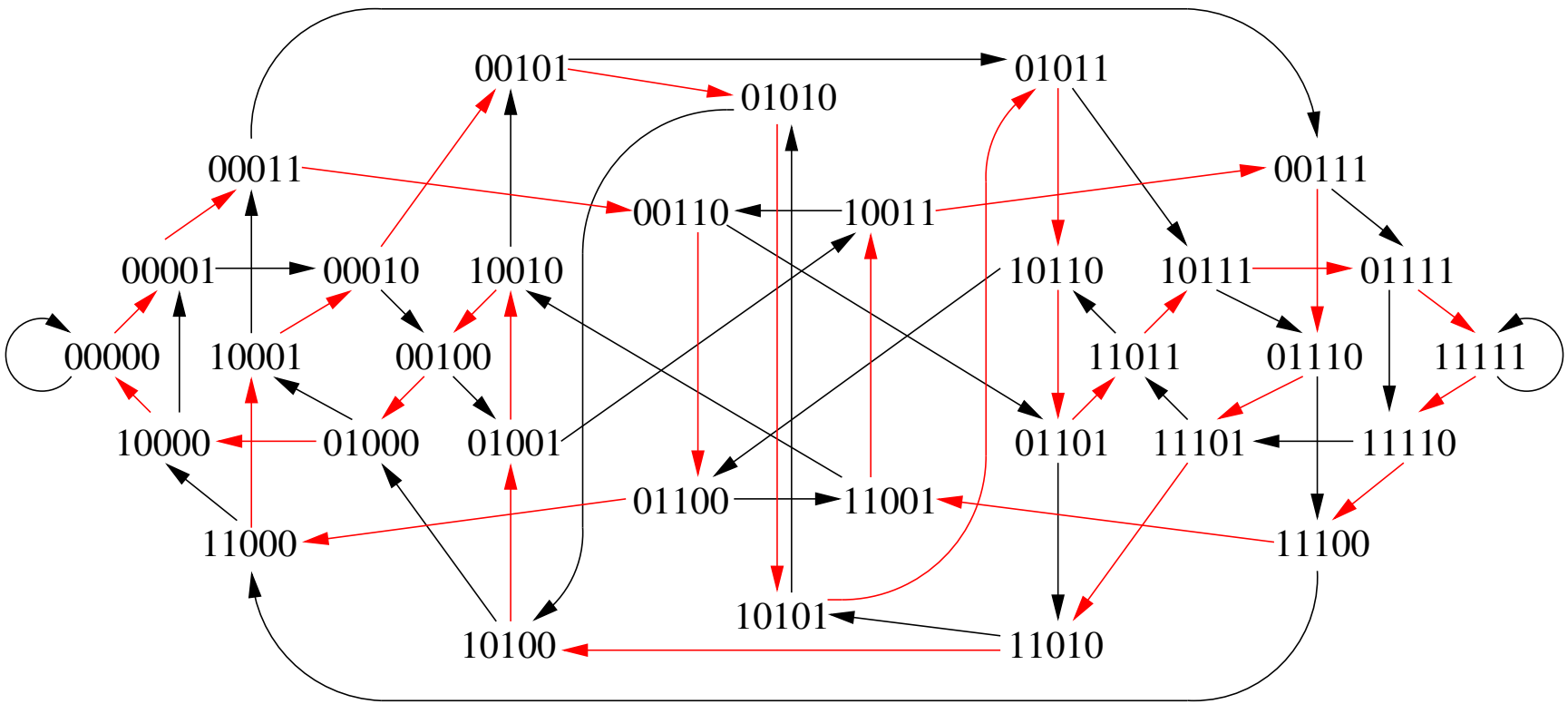
Tour de cartes – le truc

On range les cartes de telle manière que le code de chaque carte est obtenu à partir de la suivante par décalage. Cela revient à trouver un cycle hamiltonien dans le graphe de de Bruijn.



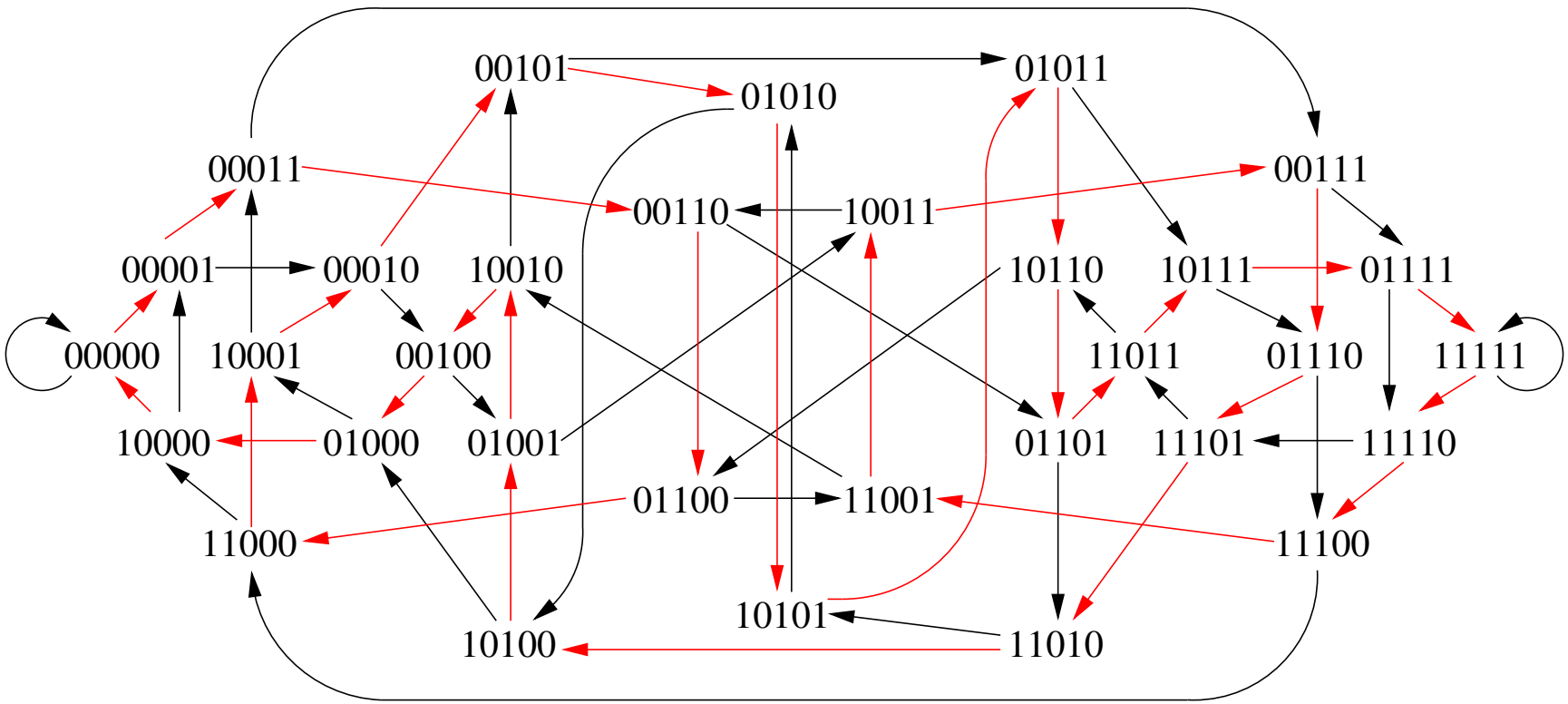
Tour de cartes – le truc

5e bit de 1e carte = 1e de la 5e carte.
5e bit de 2e carte = 2e de la 5e carte.
5e bit de 3e carte = 3e de la 5e carte.
5e bit de 4e carte = 4e de la 5e carte.
5e bit de 5e carte = 5e de la 5e carte.



Tour de cartes – le truc

Un cycle hamiltonien est
A♡-A♣-A♠-7♢-9♡-D♡-10♣-A♢-7♣-8♢-V♣-8♠-V♢-9♣-D♠-V♠-9♠-R♠-R♢-R♡-D♣-10♠-7♠-9♢-R♣-D♢-V♡-8♣-10♢-7♡-8♡-10♡



Applications des graphes de Bruijn

- **Quasi-optimum pour le problème (Δ, D).** Réseaux de faible degré et faible diamètre.

- **Séquençage de chaînes d'ADN**. Grande flexibilité pour gérer les duplications, inversions ou transpositions.