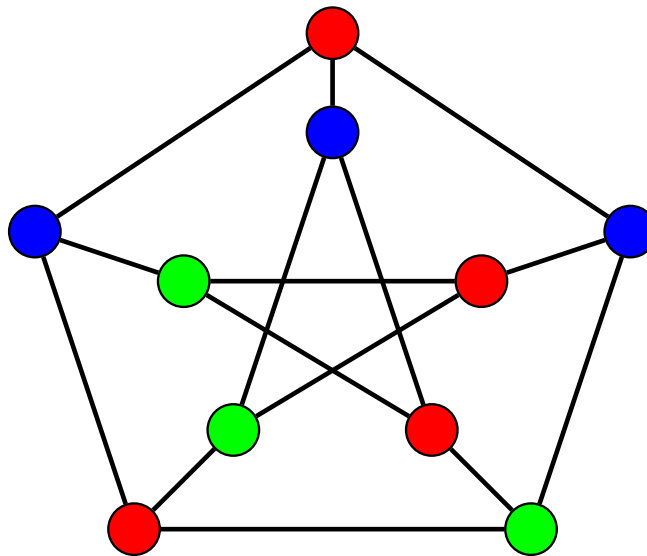


Coloriage de sommets

Coloriage de sommets

Colorer un réseaux signifie attribuer une couleur à chacun de ses noeuds de manière que deux noeuds voisins soient de couleur différente.

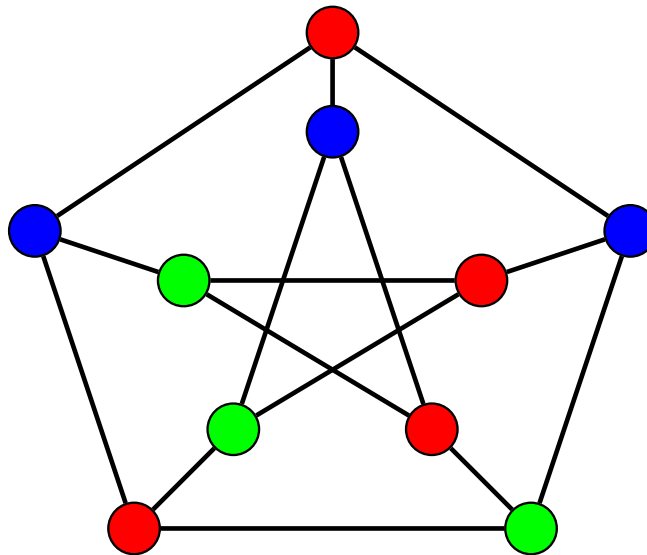


Une coloration du graphe de Petersen avec 3 couleurs

Coloriage de sommets

Définition

Soit un réseau $G = (V, E)$, soit c_v la couleur de chaque noeud $v \in V$ tel que : $\forall (v, w) \in E : c_v \neq c_w$



Une coloration du graphe de Petersen avec 3 couleurs

Applications

- La coloration de sommet a de nombreuses applications pratiques.
- Par exemple dans le domaine de réseaux sans fil où la coloration est la base de protocoles tel que MAC TDMA.
- En général, la coloration de sommets est utilisé comme un moyen pour casser des symétries, un des thèmes principaux dans le calcul distribué.

Modèle

- Chaque noeud à un identifiant unique
- Chaque identifiant peut-être codé en $\log n$ bits, où n est le nombre de noeuds.
- Sans perte de généralité, on suppose que les identifiants sont noté de 1 à n .

Nombre chromatique

Définition

Soit un graphe non orienté $G = (V, E)$, le nombre chromatique $\chi(G)$ est le nombre minimum de couleur qu'il faut pour résoudre le problème de la coloration de sommets.

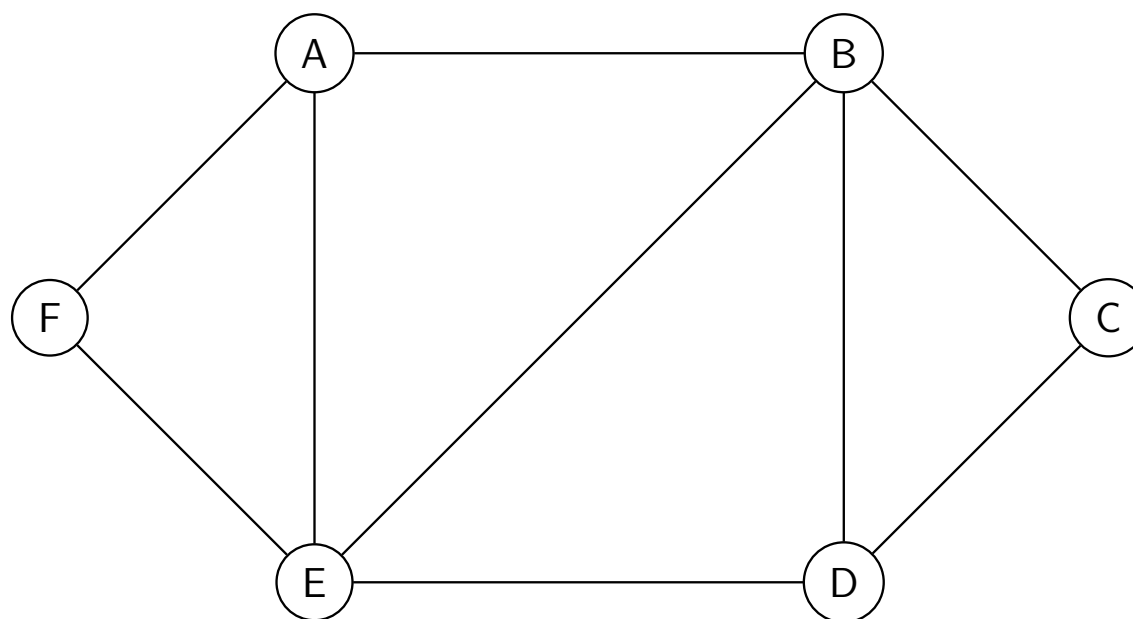
Algorithme glouton centralisé

Algorithm 1: Algorithme glouton

tant que *il existe un noeud non-coloré v* **faire**

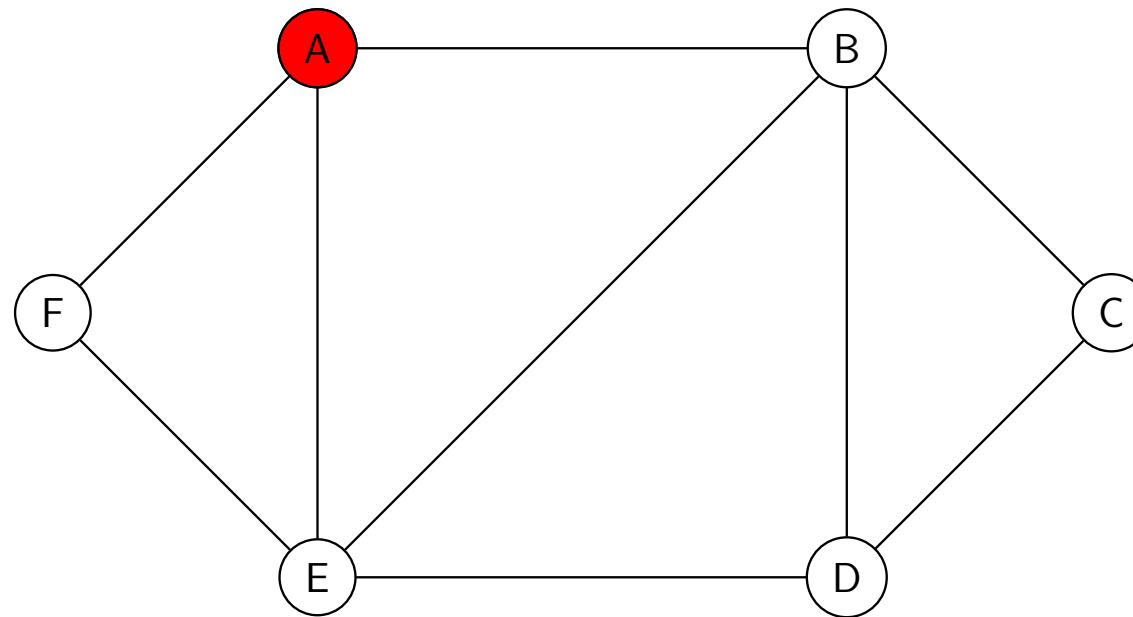
 Colorié v avec la couleur minimale qui n'entre pas en
 conflit avec ces voisins

Exemple



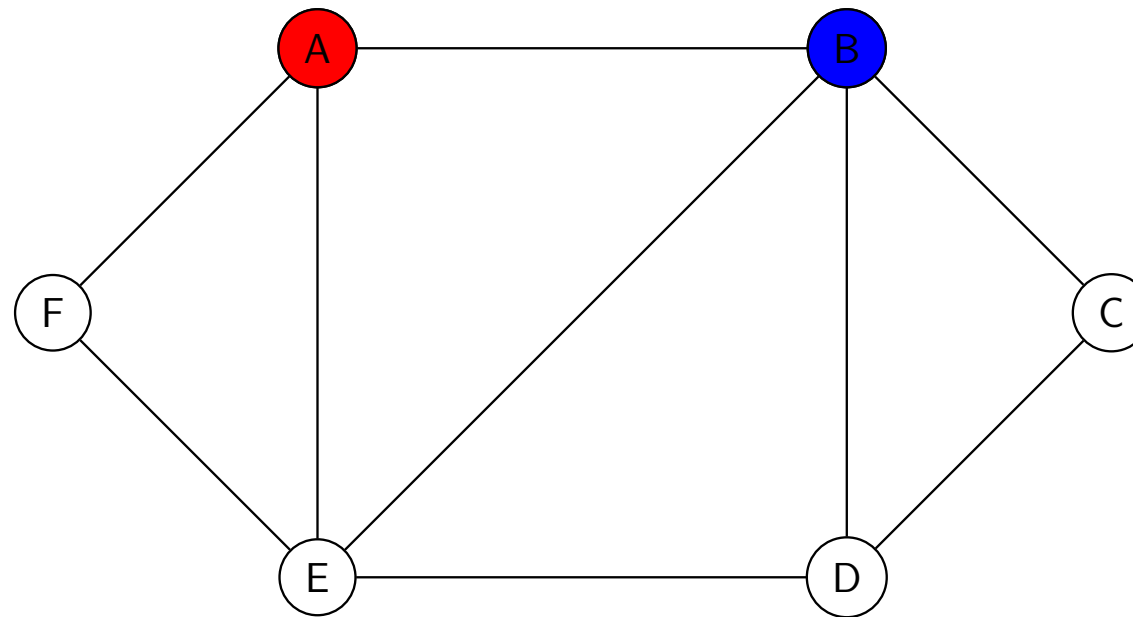
Couleurs : 1 2 3 4

Exemple



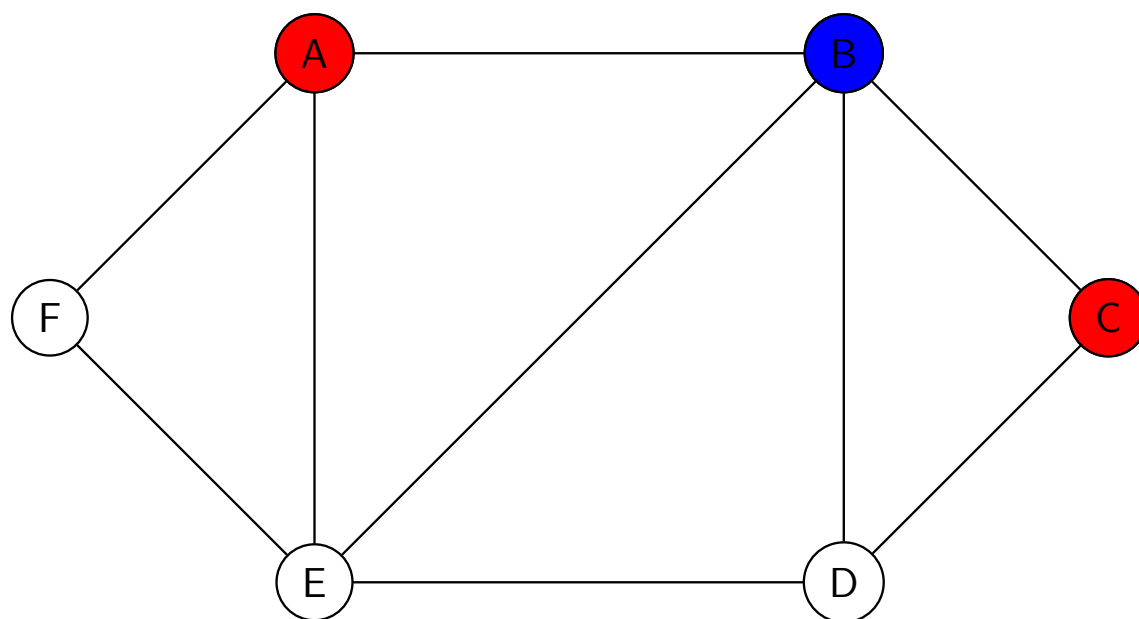
Couleurs : 1 2 3 4

Exemple



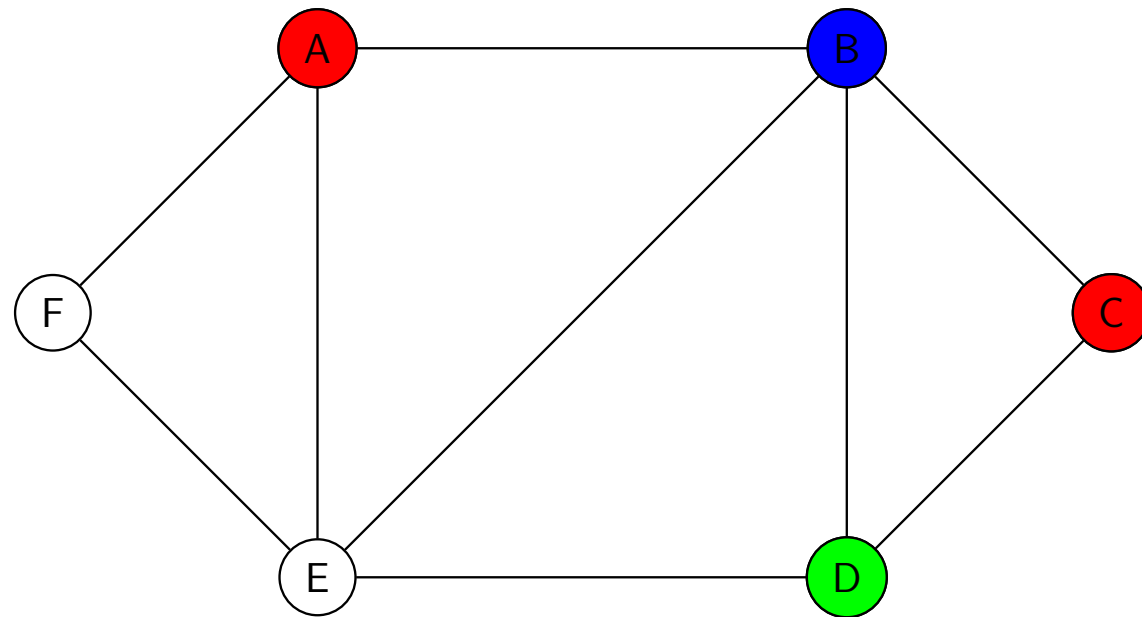
Couleurs : 1 2 3 4

Exemple



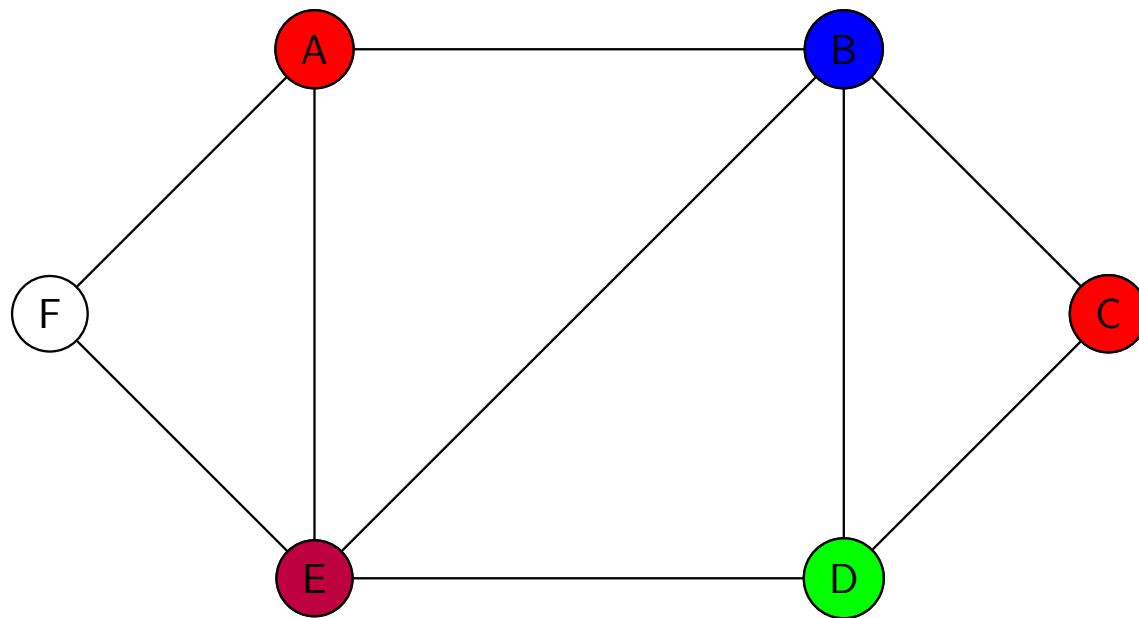
Couleurs : 1 2 3 4

Exemple



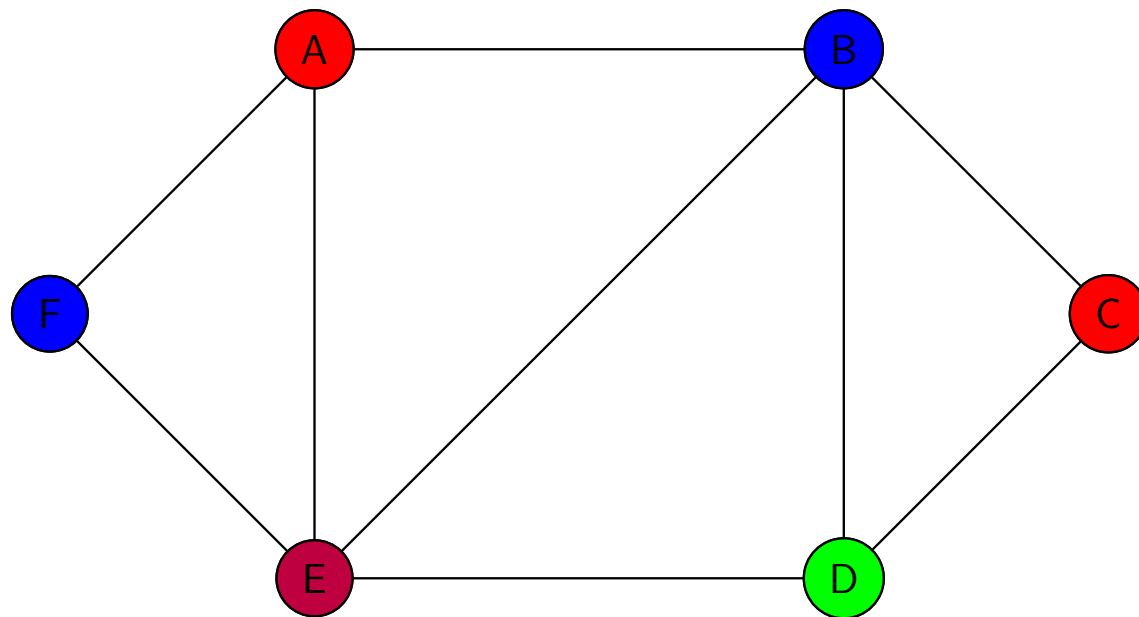
Couleurs : 1 2 3 4

Exemple



Couleurs : 1 2 3 4

Exemple

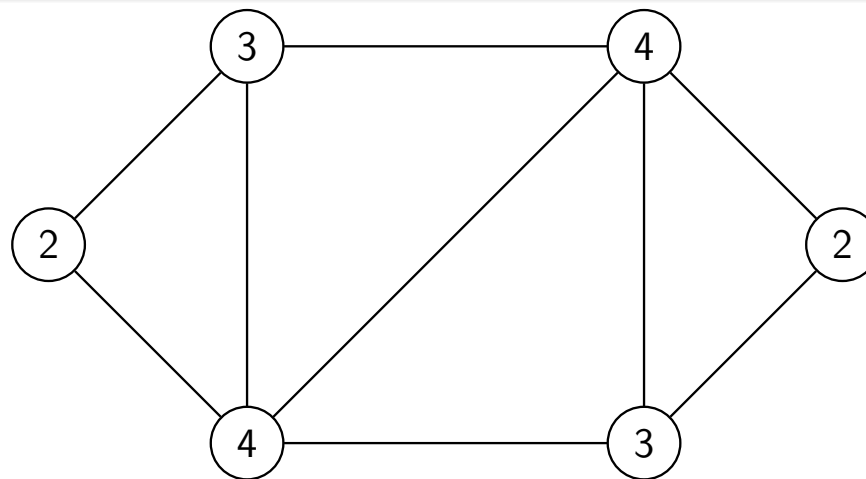


Couleurs : 1 2 3 4

Degré d'un noeud

Définition

Le nombre de voisins d'un noeud v , est noté par $d(v)$, et appelé degré de v .

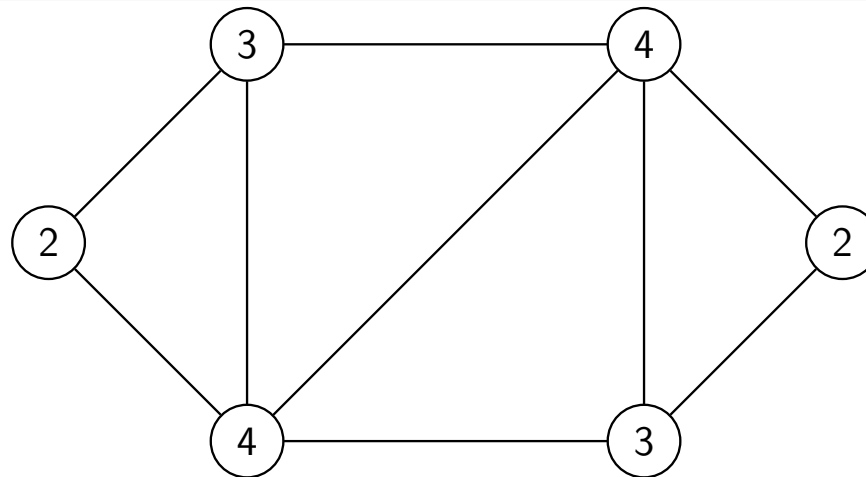


Degrés des noeuds

Degré du graphe

Définition

Le degré du graphe est le degré maximum de tout les noeuds du graphe, on le note $\Delta(G)$ ou simplement Δ .



Graphe de degré 4

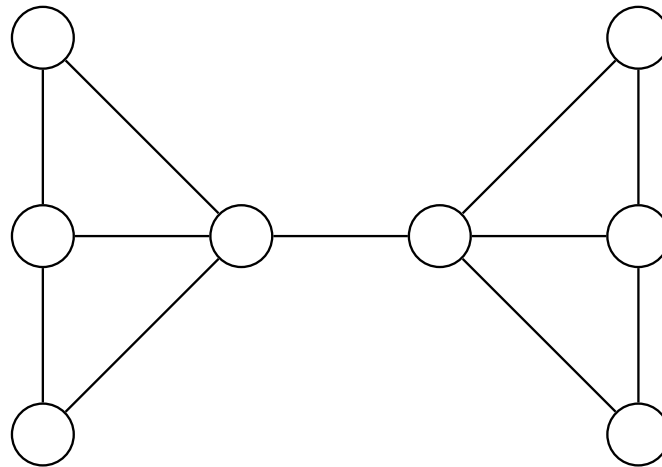
Théorème

L'algorithme glouton centralisé est correct et termine en n étapes. Il utilise $\Delta + 1$ couleurs.

Algorithme glouton distribué

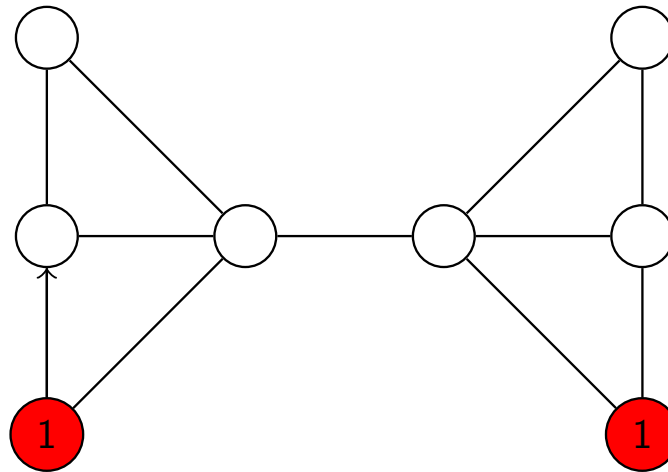
- **Initialisation de v :**
 - $Etat_v := Reveille$
 - $c_v := \emptyset$
 - $\forall u \in N(v) : c_u := \emptyset$
- **Procédure ChoixCouleur :**
 - $couleur := \{1, \dots, \Delta\}$
 - $\forall u \in N(v) : couleur := couleur - \{c_u\}$
 - $c_v := \min\{couleur\}$
 - $\forall u \in N(v) : \text{Envoie } \langle Color, c_v \rangle$
- **Réveil spontané :**
 - Initialisation
 - ChoixCouleur
- **Lors de la réception de $\langle Color, c \rangle$ envoyé par u :**
 - Si $Etat_v \neq Reveille$ faire **Initialisation**
 - $c_u := c$
 - **ChoixCouleur**

Approche distribuée : vision locale



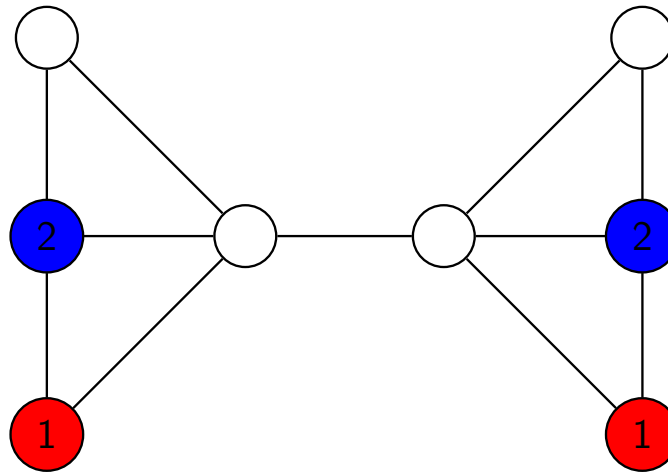
Couleurs : 1 2 3 4 5

Approche distribuée : vision locale



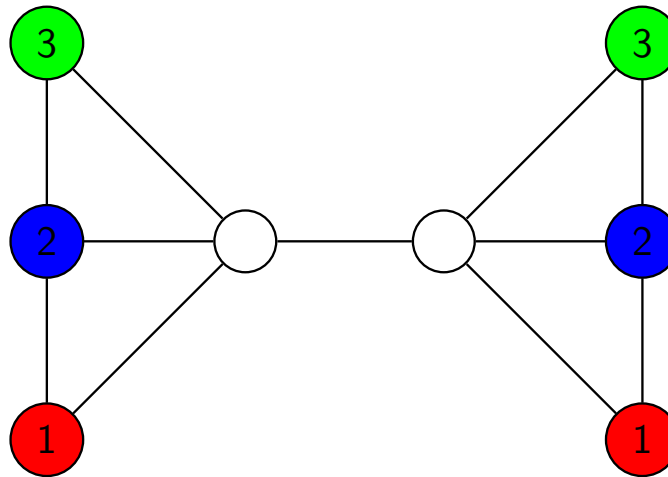
Couleurs : 1 2 3 4 5

Approche distribuée : vision locale



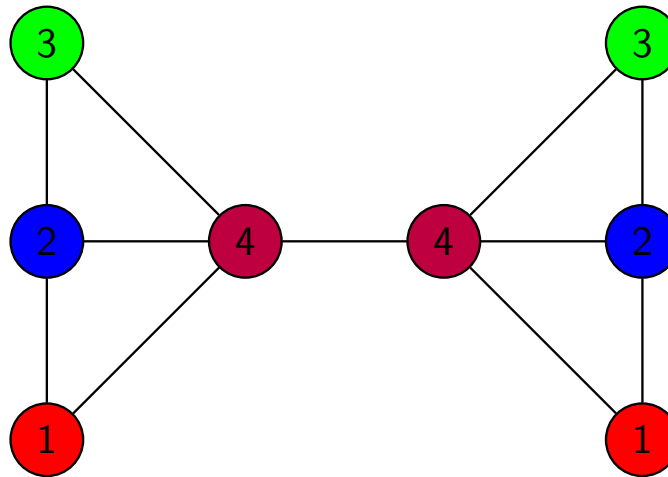
Couleurs : 1 2 3 4 5

Approche distribuée : vision locale



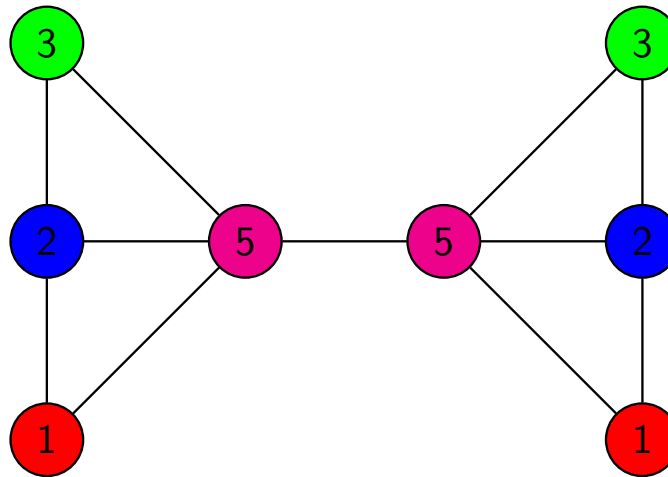
Couleurs :     

Approche distribuée : vision locale



Couleurs :     

Approche distribuée : vision locale



Couleurs : 1 2 3 4 5

Question :

Comment éviter ce problème ?

Question :

Comment éviter ce problème ?

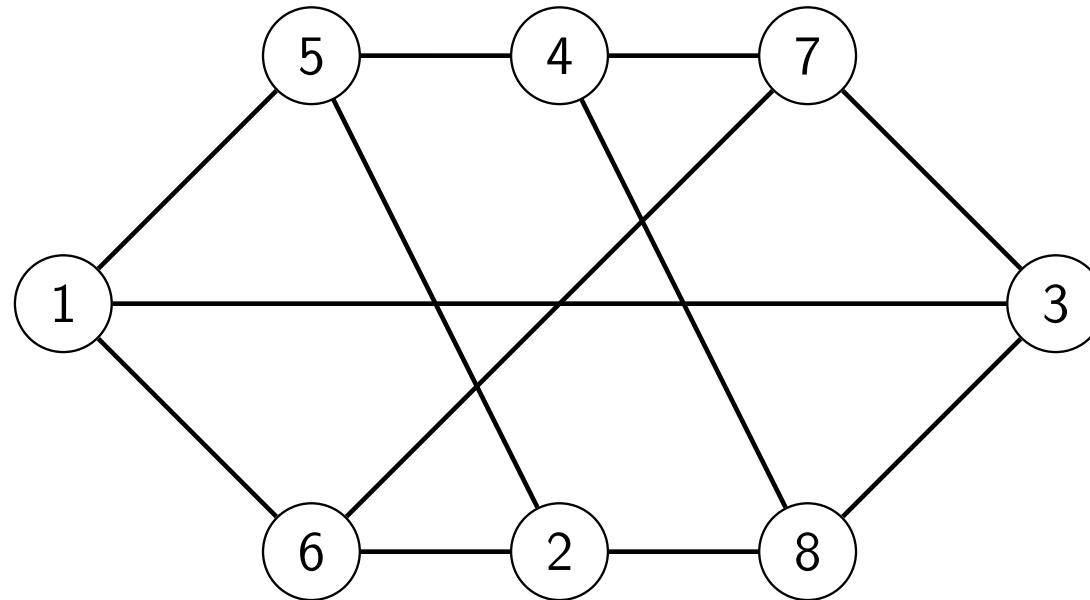
Réponse :

Utiliser les identifiants.

Algorithme glouton distribué

- **Initialisation de v :**
 - $Etat_v := Reveille$
 - $c_v := \emptyset$
 - $\forall u \in N(v) : c_u := \emptyset$ et Envoie $\langle Color, \emptyset \rangle$ à u
- **Procédure ChoixCouleur :**
 - Si $\forall u \in N(v) \mid id(u) > id(v) \wedge c_u \neq \emptyset$
 - $couleur := \{1, \dots, \Delta\}$
 - $\forall u \in N(v) : couleur := couleur - \{c_u\}$
 - $c_v := \min\{couleur\}$
 - $\forall u \in N(v) : \text{Envoie} \langle Color, c_v \rangle$
- **Réveil spontané :**
 - Initialisation
 - ChoixCouleur
- **Lors de la réception de $\langle Color, c \rangle$ envoyé par u :**
 - Si $Etat_v \neq Reveille$ faire **Initialisation**
 - $C_u := c$
 - **ChoixCouleur**

Exemple : Le noeud 1 se réveille.



Couleurs : 1 2 3 4 5

Arbres

Question :

Combien de couleur faut-il pour colorier un arbre ?

Arbres

Question :

Combien de couleur faut-il pour colorier un arbre ?

Réponse :

- Il faut 2 couleurs.
- Comment faire ?

LEMME

Lemme I

Le nombre chromatique d'un arbre est inférieur ou égal à 2.

Preuve (constructive)

Si la distance du noeud à la racine est paire, la couleur est 1, sinon (impair) la couleur est 0. Les noeuds à une distance pair ont uniquement des voisins à une distance impair et vice versa. Si on suppose que chaque noeud connaît son parent (la racine n'a pas de parent) et ses enfants dans l'arbre, cette preuve donne l'algorithme suivant.

ALGORITHME LENT DE COLORIAGE D'ARBRE

■ Initialisation de la racine:

$\text{couleur}_r = 0$

envoyer $\langle \text{Couleur}, 0 \rangle$ à tous ses enfants

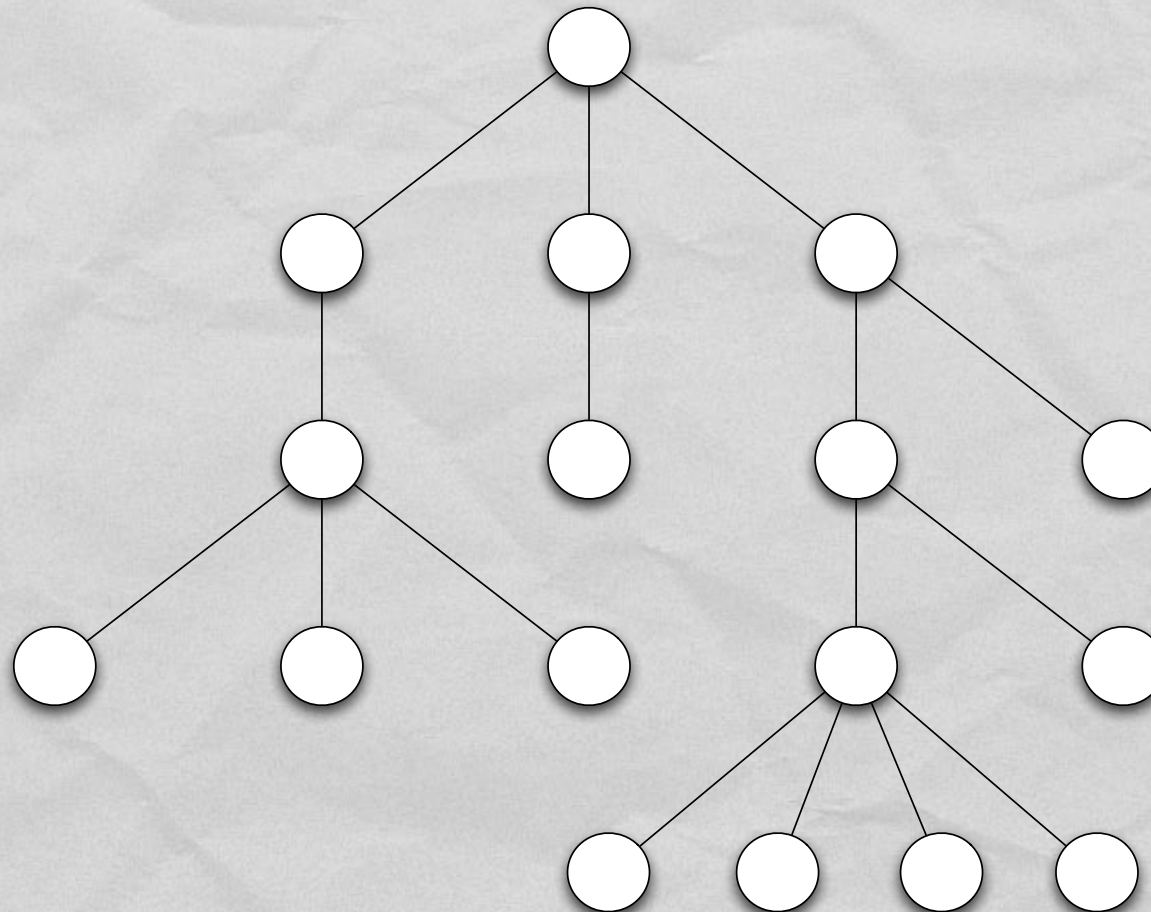
ALGORITHME LENT DE COLORIAGE D'ARBRE

■ Lors de la réception de $\langle \text{Color}, c \rangle$ de parent_v :

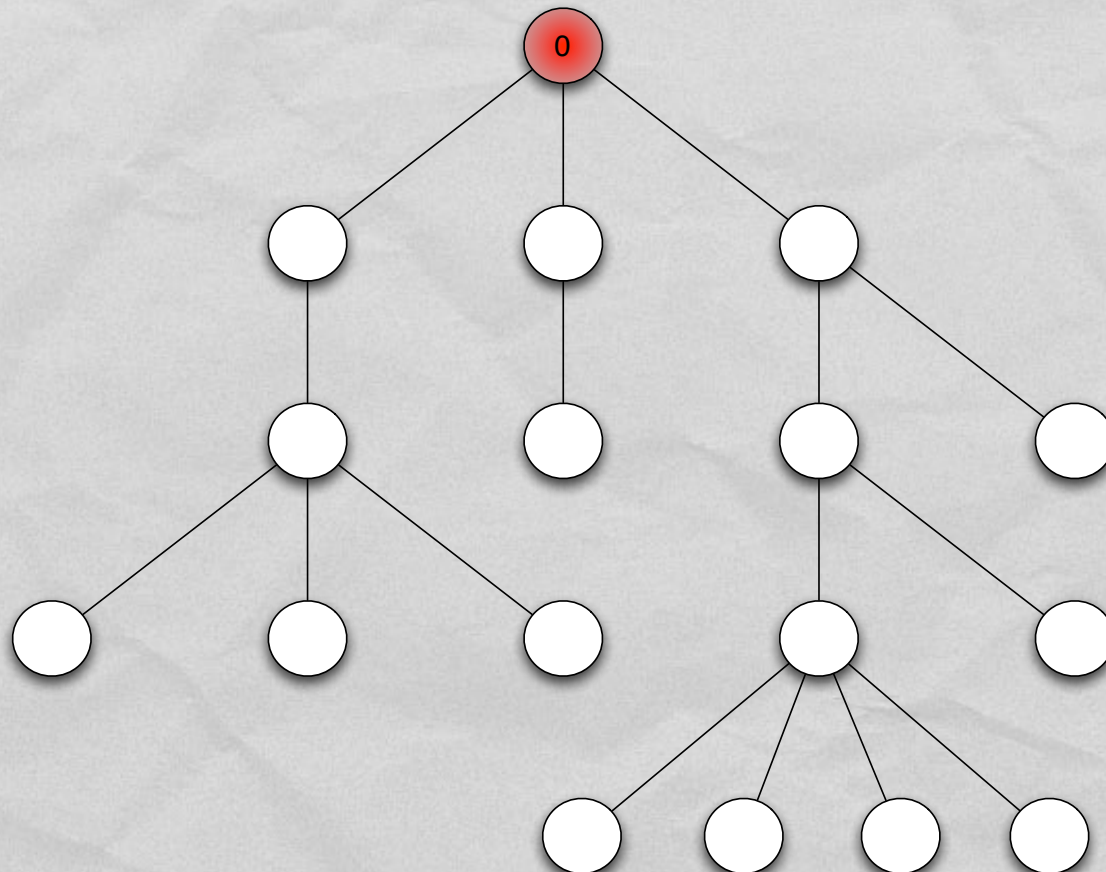
$\text{couleur}_v = c - 1$

envoyer $\langle \text{Couleur}, \text{couleur}_v \rangle$ à tous ses enfants

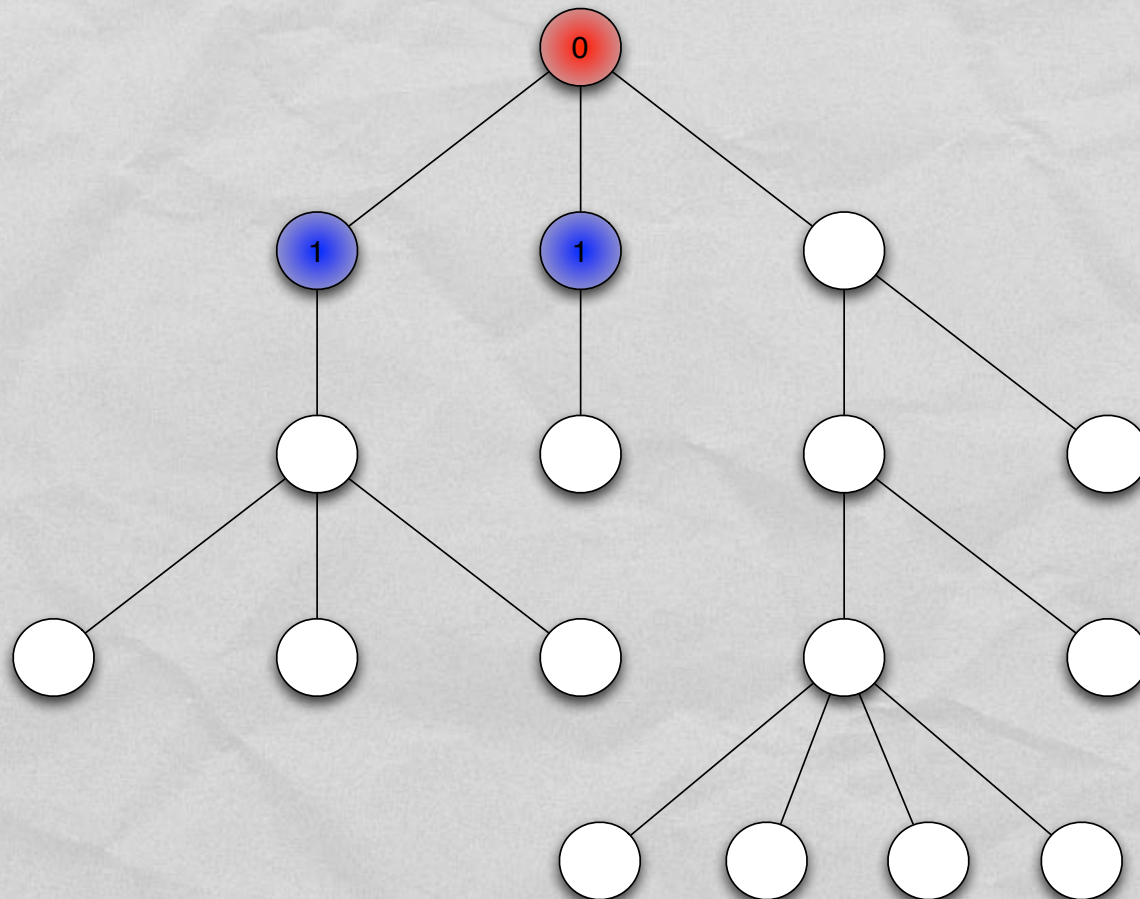
EXEMPLE



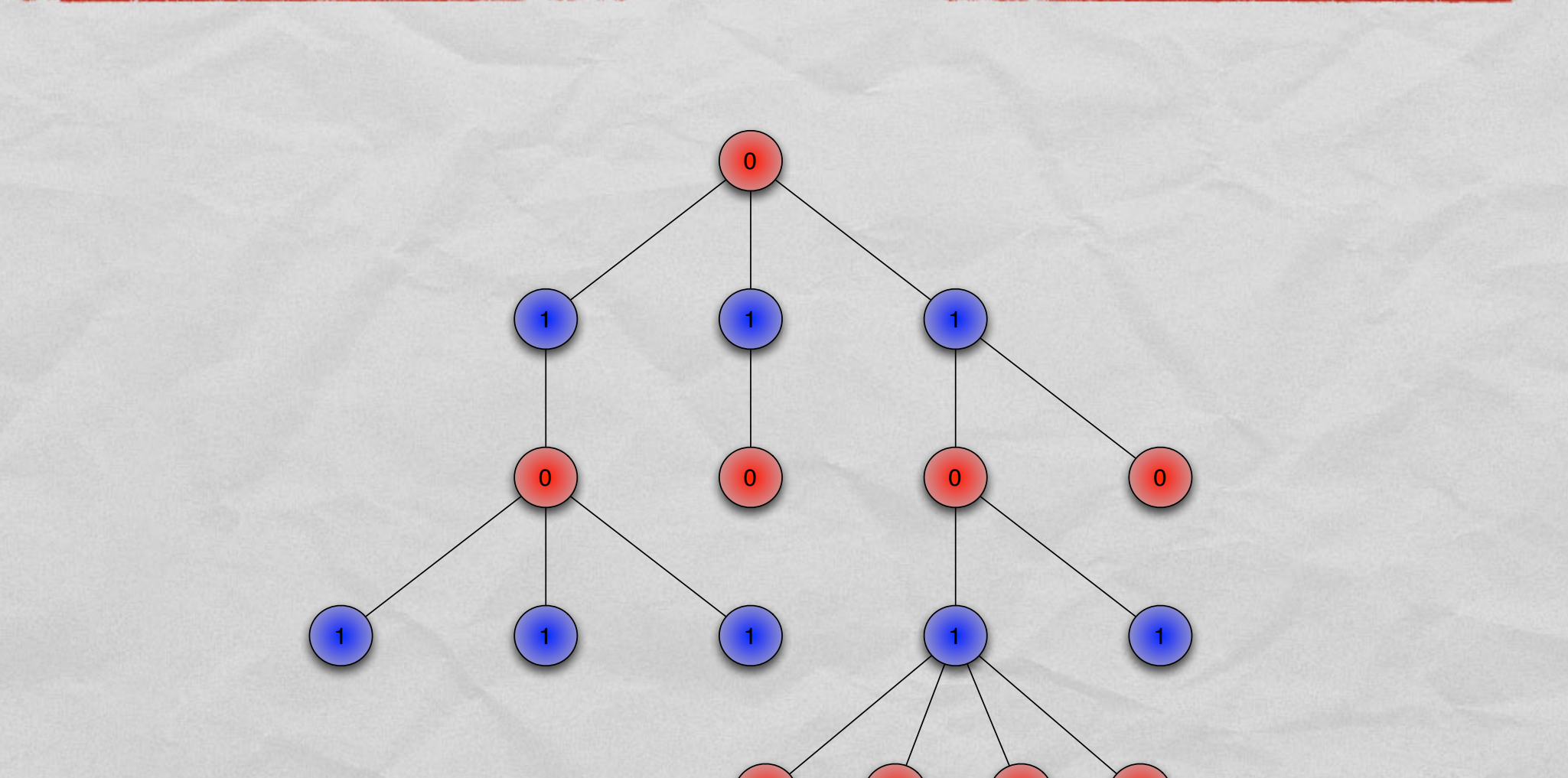
EXEMPLE



EXEMPLE



EXAMPLE



REMARQUES

- Pour la preuve de l'algorithme on utilise le lemme 1
- Cette algorithme fonctionne si une racine est désignée.
- La complexité temporelle:
 - égale à la hauteur de l'arbre
 - $O(\log n)$
- Algorithme synchrone ou asynchrone?

ALGORITHMES ASYNCHRONES

Définition 6

En asynchrone, l'algorithme fonctionne grâce à la réception et envoie de message. Les processus non pas d'horloge, de mémoire communes. Un message envoyer par un processeur à un autre processeur finira par arriver en un temps fini mais non borné.

REMARQUES

- Les modèles synchrones et asynchrones sont les fondements de l'algorithmique distribuée (répartie)
- Il existe d'autres modèles.

ALGORITHMES ASYNCHRONES

COMPLEXITÉ TEMPORELLE

Définition 7

Pour les algorithmes asynchrones la complexité temporelle est le nombre d'unité de temps du début de à la fin de l'exécution dans son pire des cas. Chaque transmission de message est compté comme une unité de temps.

ALGORITHMES ASYNCHRONES

COMPLEXITÉ EN MESSAGES

Définition 8

La complexité en message pour un algorithme synchrone ou asynchrone est déterminé par le nombre de message échangé dans la «pire» des exécutions.

ALGORITHMES ASYNCHRONES

COMPLEXITÉ EN MESSAGES

Théorème 3

L'algorithme «lent de coloriage» est correct. Si chaque noeud connaît son parent et ses enfants. Le temps (asynchrone) est la hauteur de l'arbre qui est borné par le diamètre de l'arbre. La complexité en message est de $n-1$ dans un arbre à n noeuds.

REMARQUES

- Pour cette algorithme la complexité temporelle est la même dans le cas asynchrone et synchrone.
- Existe t-il un algorithme plus efficaces?

ALGORITHMES ASYNCHRONES

COMPLEXITÉ EN MESSAGES

Définition 9 (Log-étoile)

$$\forall x \leq 2: \log^* x := 1$$

$$\forall x > 2: \log^* x := 1 + \log^*(\log x)$$

REMARQUES

- Log-étoile est une fonction qui croît de façon «incroyablement» lentement.
- Log-étoile du nombre d'atomes observable dans l'univers (estimé à 10^{80}) est de 5.

ALGORITHME 6-COULEURS

- Topologie: arbre
- Modèle: synchrone

ALGORITHME 6-COULEURS

■ Pour chaque site v :

$c_v := \text{id}_v(\text{en binaire})$

$L_v := |c_v|$

si v est la racine

$L_v := 0$

envoie $\langle c_v \rangle$ à ses enfants

ALGORITHME 6-COULEURS

- Réception de $\langle c_p \rangle$ du parent:

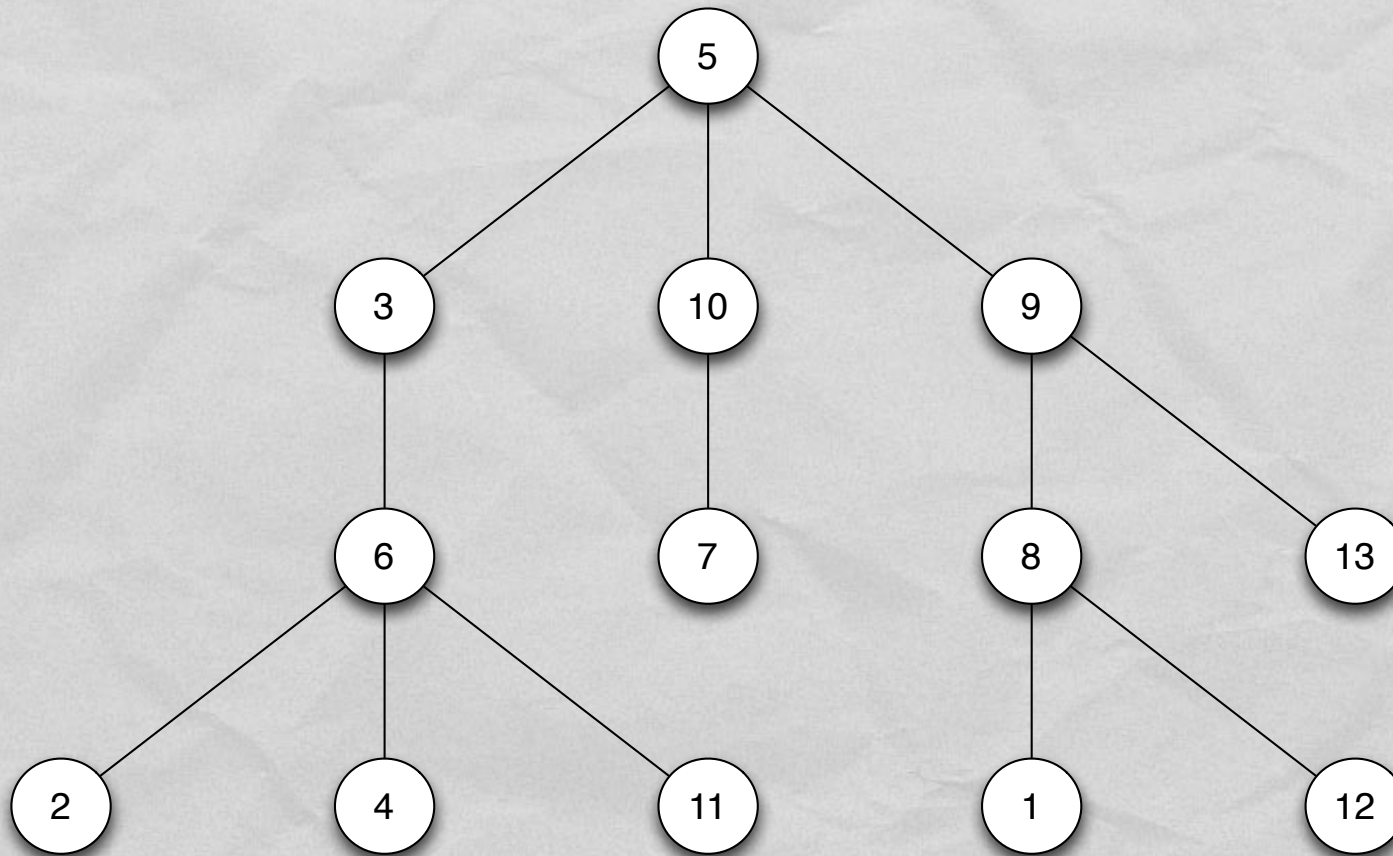
Jusqu'à ce que $|c_v| \neq 6$

$L_v := \min\{i \mid c_v[i] \neq c_p[i]\}$

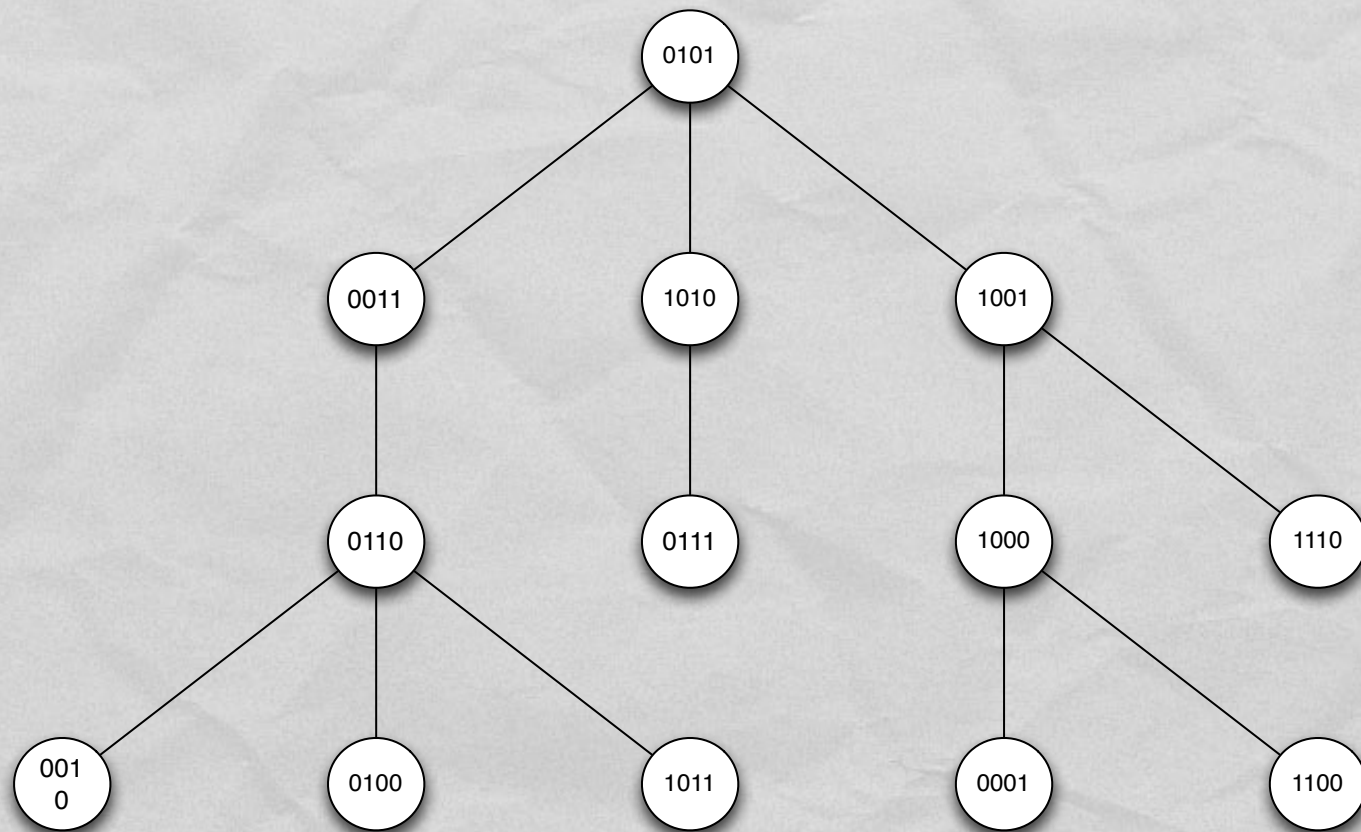
$c_v := (L_v; c_v[L_v])$

envoie $\langle c_v \rangle$ aux enfants

EXEMPLE



EXEMPLE





Théorème 4

L'algorithme 6-couleurs se termine en $\log^*n$ étapes.

MODÈLE

- Pour pouvoir comparer des programmes réparties il faut savoir si ils fonctionnent sur le même modèle
- A savoir
 - système synchrone ou asynchrone
 - topologie du réseau:
 - arbitraire, complet, arbre, cycle...
 - Information globale ou pas
 - Nombre de noeuds du réseau, diamètre, degré...

REMARQUES

- La couleurs 11^* (en notation binaire, c'est à dire 6 ou 7 en notation décimale) ne sera pas choisi

BIBLIOGRAPHIE

- La partie de ce cours sur la coloration est basé sur le cours de:
 - Roger Wattenhofer
 - intitulé «Principles of Distributed Computing»

TOPOLOGIE

- Bien sur un algorithme est plus efficace si il fonctionne sur n'importe quel graphe
- Mais les algorithmes sur une topologie particulière peuvent servir
 - on utilisera pour cela les réseaux virtuels
 - c'est à dire une topologie particulière crée grâce par exemple aux tables de routages sur le réseau réel

DYNAMISME / PANNES

- La réalité des réseaux c'est leur grande «mobilité»
 - cette mobilité ou dynamisme est dû
 - au départ ou arrivée de machines
 - aux pannes
 - aux malveillances
- Avant de savoir gérer ce dynamisme
 - il faut déjà réussir à gérer des réseaux statiques
 - c'est le but de ce cours

Modèle

- ❖ Le réseau modélisé par un graphe $G = (V, E)$
 - ❖ Les sommets (ou nœuds) modélisent les entités de calcul,
 - ❖ Les arêtes modélisent les liens de communications.
- ❖ Chaque nœud possède un identifiant distinct entre 1 et $n = |V|$.
 - ❖ On note $\text{Id}(u)$ l'identité du nœud u .

Modèle synchrone

- ❖ Dans modèle synchrone, les calculs s'effectuent par étapes synchrones.
- ❖ A chaque étape, chaque nœud u effectue trois opérations :
 1. Envoyer un message à un ou plusieurs voisins de u dans G ;
 2. Recevoir les messages des des voisins de u dans G ;
 3. Effectuer un calcul interne (facultatif).

Estampilles

- ❖ Chaque message peut être estampillé par l'identité de son expéditeur.

Coloration

- ❖ Une k -coloration d'un graphe $G = (V, E)$ est une fonction

$$c : V \rightarrow \{1, \dots, k\}$$

telle que, pour tout arête $\{u, v\} \in E$, $c(u) \neq c(v)$.

- ❖ Le plus petit k pour lequel il existe une k -coloration de G est appelé le nombre chromatique de G , noté $\chi(G)$.

Remarques

- ❖ Etant donné un graphe G et un entier k , décider si $\chi(G) \leq k$ est np-complet.
- ❖ Approximer $\chi(G)$ est également très difficile : pour tout $\epsilon > 0$, sauf si $p = np$.
- ❖ En revanche, tout graphe admet une $(\Delta + 1)$ -coloration,
 - ❖ $\Delta = \Delta(G)$ dénote le degré maximum des sommets de G :

$$\Delta = \max_{u \in V} \deg_G(u)$$

Algorithme glouton

- **début**
 - $C = \emptyset, \text{Color} = \{1, \dots, \Delta + 1\}$
 - **Tant que** $C \neq V$ **faire**
 - choisir $u \in V \setminus C$
 - $C \leftarrow C \cup \{u\}$
 - $c(u) = \min\{\text{Color} \setminus \{c(v) \mid v \in N(u)\}\}$
- **fin**

Arguments de preuve

- ❖ L'algorithme ci-dessus retourne une $(\Delta+1)$ -coloration de G car, puisque le degré de tout sommet u satisfait $\deg(u) \leq \Delta$, chaque sommet u possède au plus Δ voisins dans G .
- ❖ En conséquence, au moins une couleur parmi les $\Delta+1$ couleurs autorisées est systématiquement disponible lorsque le sommet u est colorié par l'algorithme.

$(\Delta + 1)$ coloration distribué

Le problème de $(\Delta + 1)$ -COLORATION distribuée :

Entrée : un réseau $G = (V, E)$ de n nœuds, dont chaque nœud dispose d'une identité distincte dans $\{1, \dots, n\}$;

Objectif : chaque nœud $u \in V$ retourne une valeur $c(u) \in \{1, \dots, \Delta + 1\}$ telle que $c(u) \neq c(v)$ pour tout $\{u, v\} \in E$, où Δ dénote le degré maximum des sommets de G .

Remarques

- ❖ Ce problème semble a priori simple à résoudre:
 - ❖ Il suffit que chaque nœud se mette d'accord avec ses voisins afin de ne pas choisir une même couleur que l'un d'entre eux.
- ❖ En fait, quelque soit le choix des couleurs effectués par ses voisins, un nœud reste libre de choisir au moins une couleur.

3-coloration de l'anneau

- ❖ Variante d'un algorithme dû à Cole et Vishkin.
- ❖ Suppositions:
 - ❖ Les nœuds de l'anneau ont une notion commune de droite et de gauche
 - ❖ Les noeuds de l'anneau connaissent n .
- ❖ Notations:
 - ❖ La couleur courante d'un nœud u est notée $c(u)$
 - ❖ $c(u)$ est écrit en binaire: $c(u) = c_1(u)c_2(u) \dots c_r(u)$.
 - ❖ On note $|c(u)|$ la longueur de cette chaîne binaire.
 - ❖ $(c_1(u), k)$ correspond à un bit est la représentation binaire de k

Variante - Cole et Vishkin

❖ Algorithme pour le noeud u

début

$c(u) \leftarrow \text{Id}(u)$

tant que $|c(u)| > 3$ **faire**

envoyer $c(u)$ au voisin de droite w

recevoir $c(v)$ du voisin de gauche v

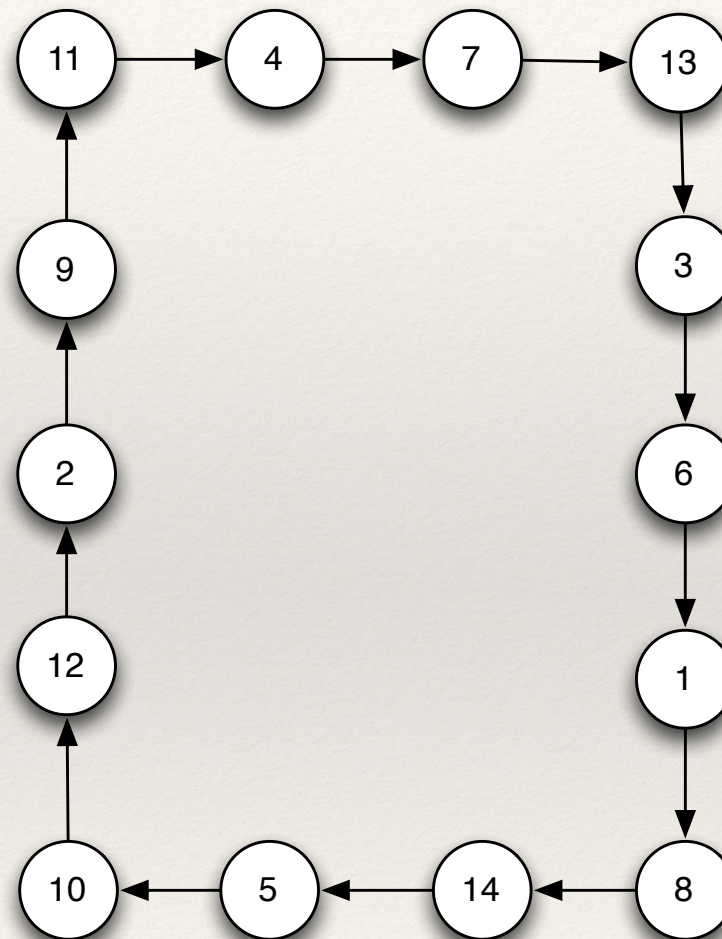
$k \leftarrow$ plus petit indice tel que $c_k(v) \neq c_k(u)$

$c(u) \leftarrow (c_k(u), k)$

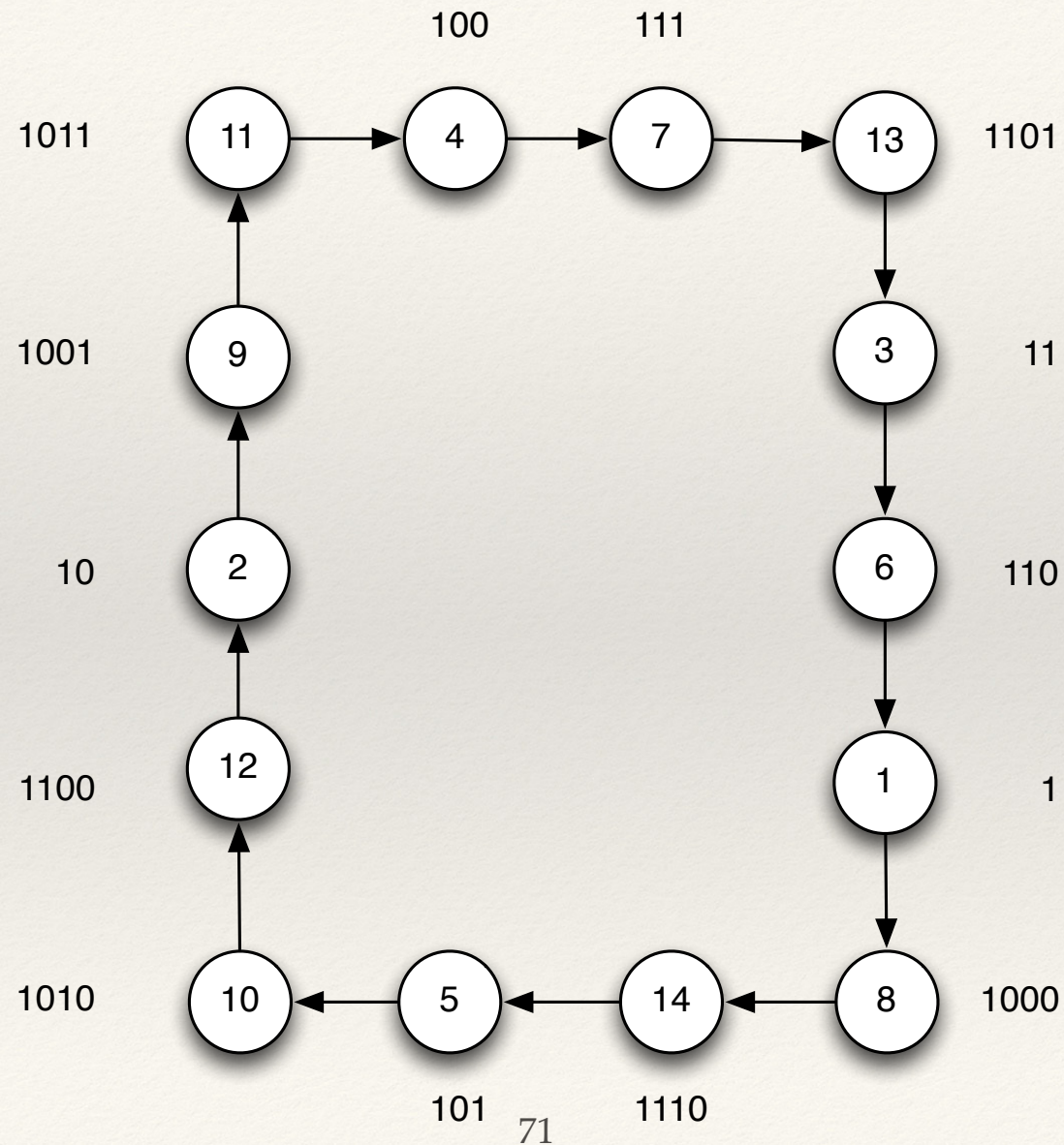
réduction-couleur()

fin.

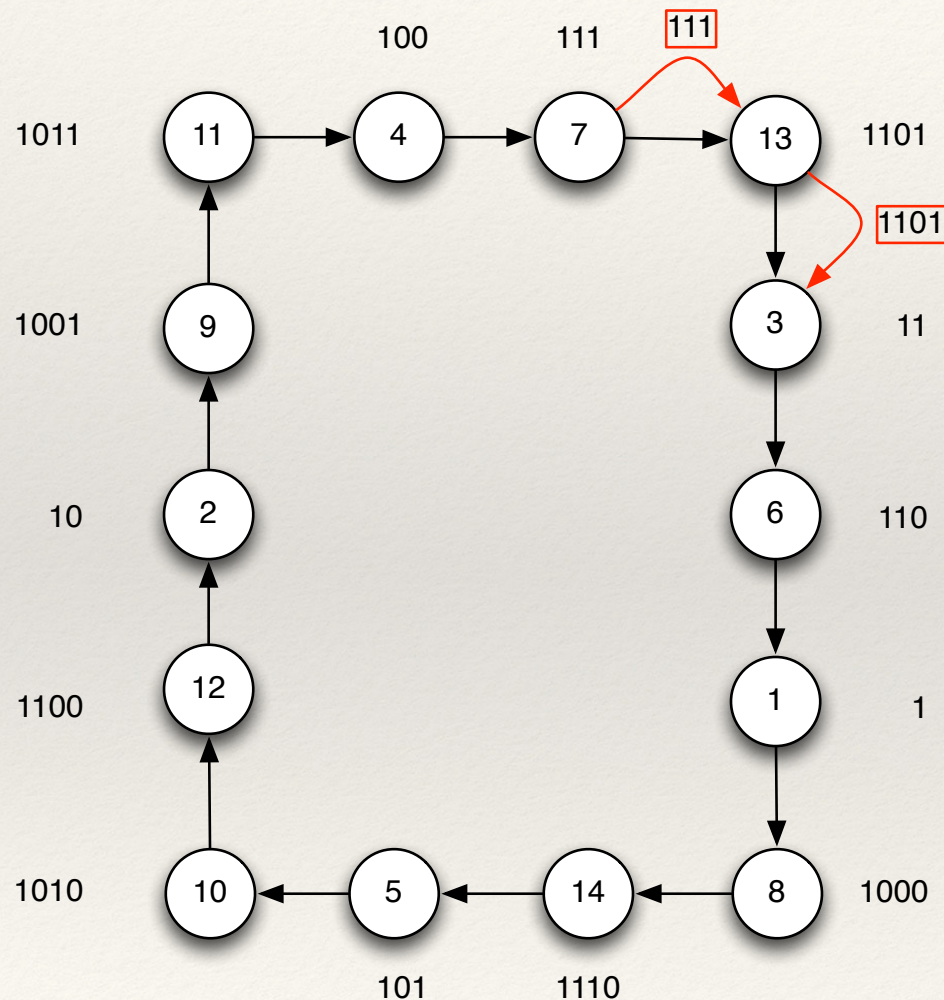
Exemple (sans réduction couleur)



Représentation binaire

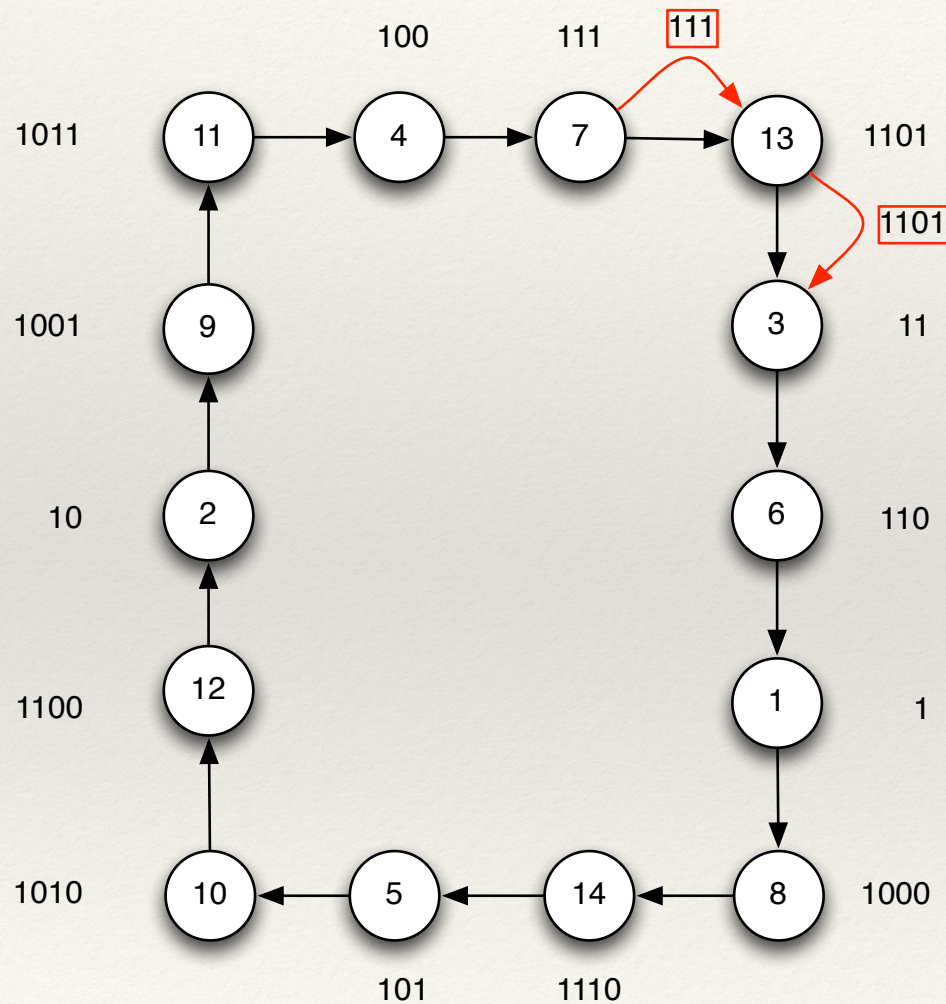


Exemple



- ❖ Le noeud 13 envoie: 1101
- ❖ Le noeud 13 reçoit: 111
- ❖ $c(u)=1101$ et $c(v)=111$
- ❖ $k=2$ en binaire 10
- ❖ $c(u)=c_k(u).k=010$

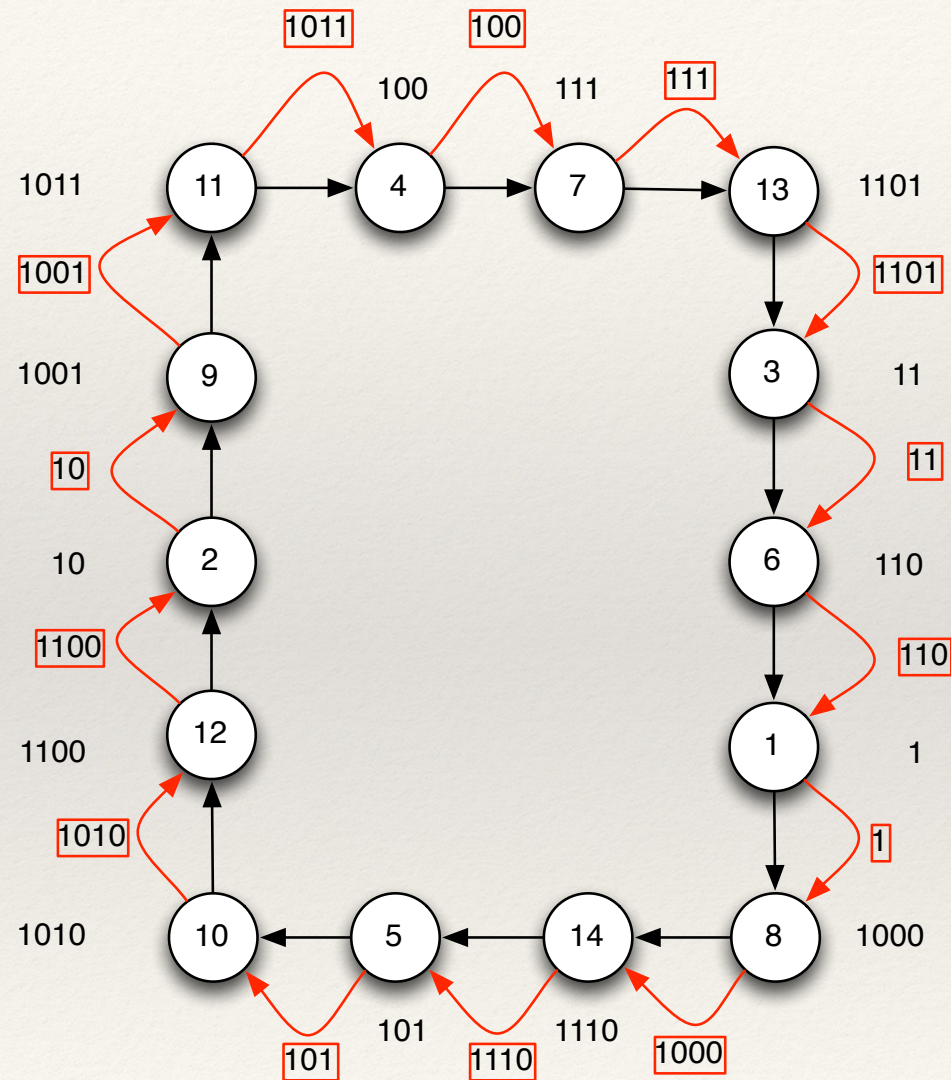
Exemple



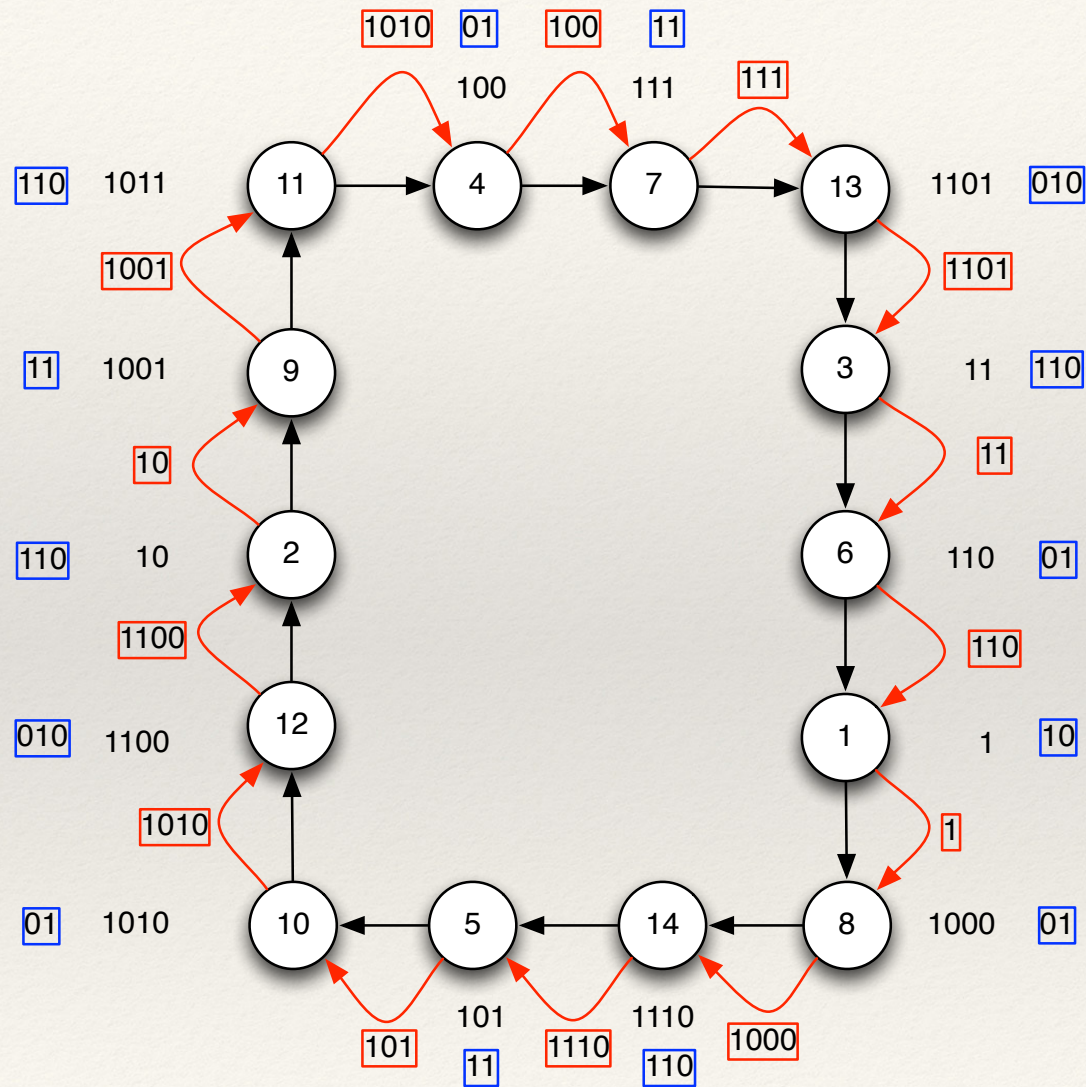
❖ Faire la même chose avec les noeuds:

❖ 11 et 8

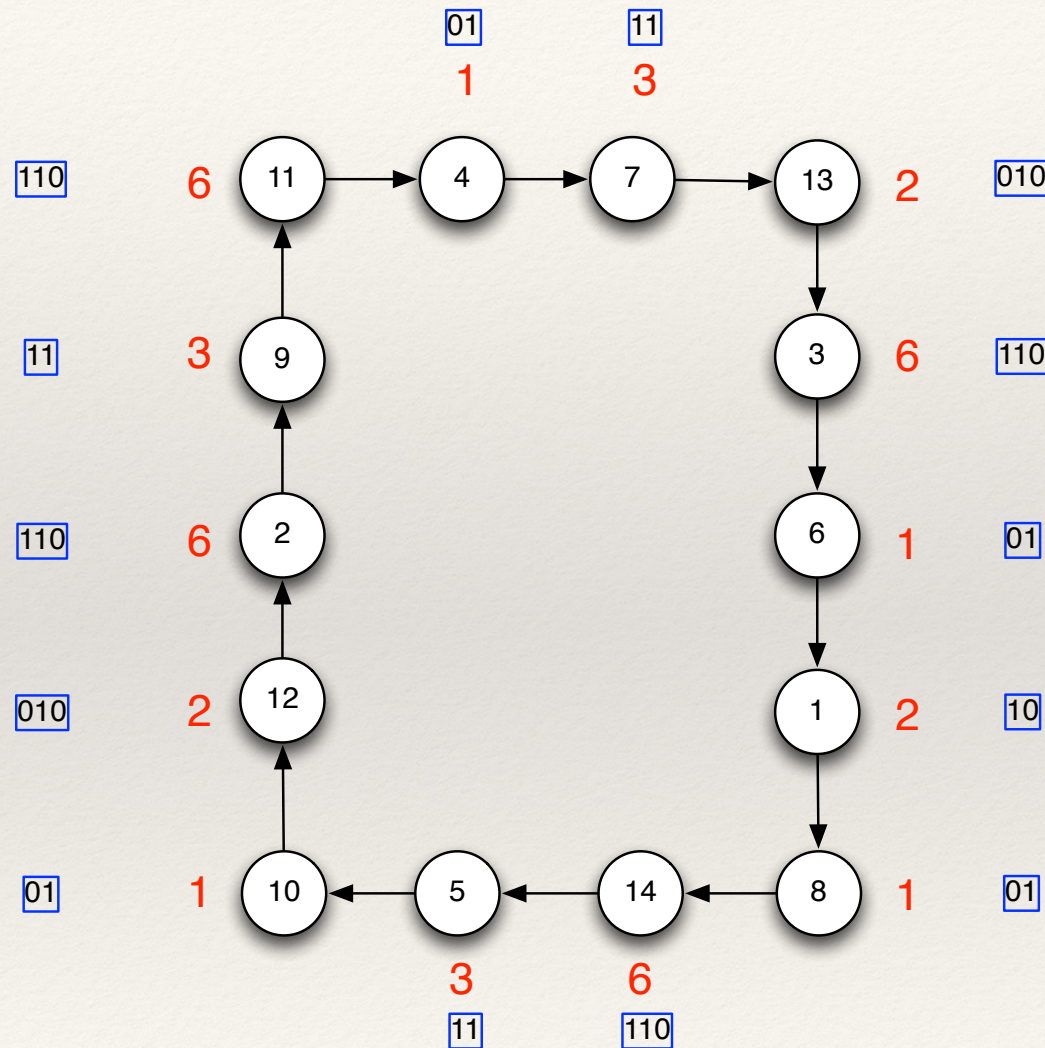
Exemple



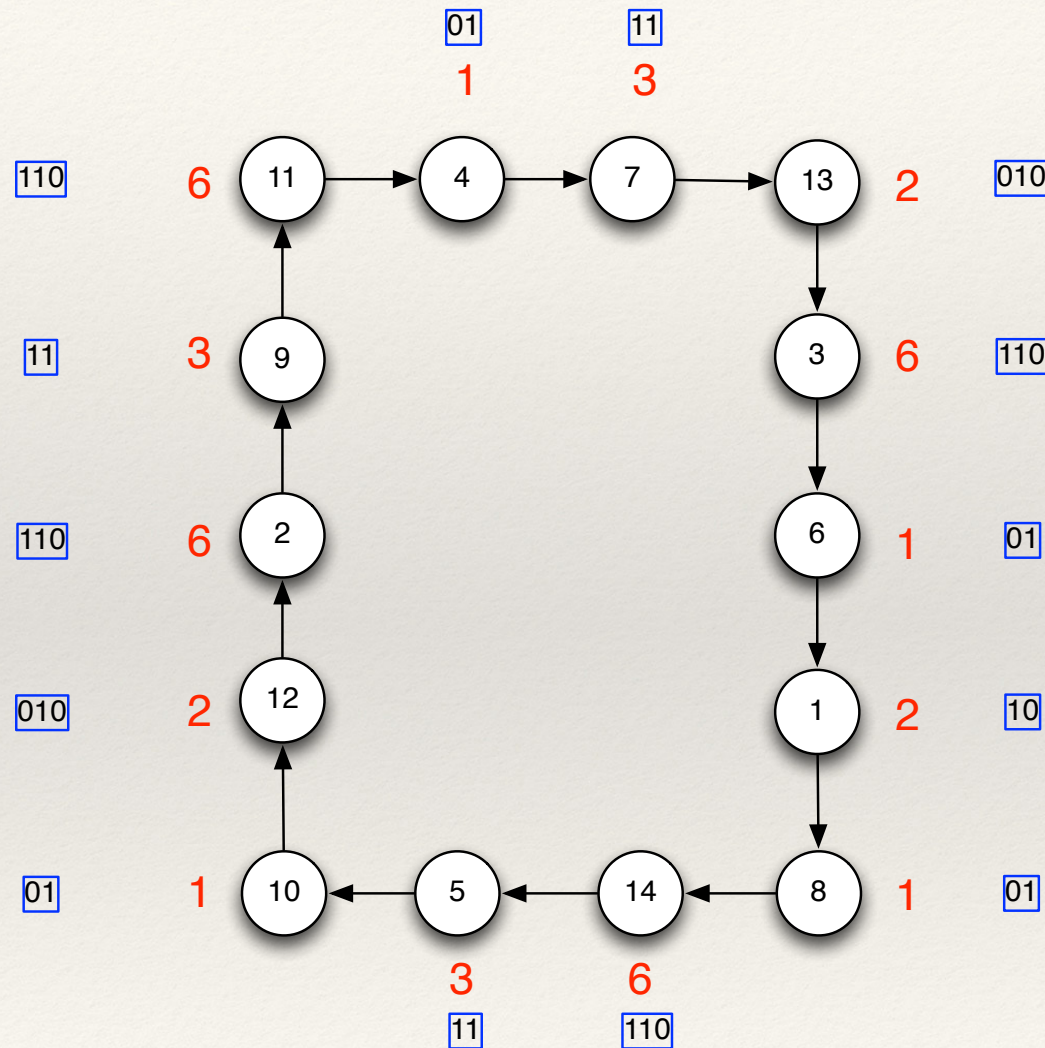
Exemple



Exemple



Exemple



Réduction de couleur

fonction réduction-couleur() **pour un sommet** u :

début

pour $i = 8$ **à** 4 **faire**

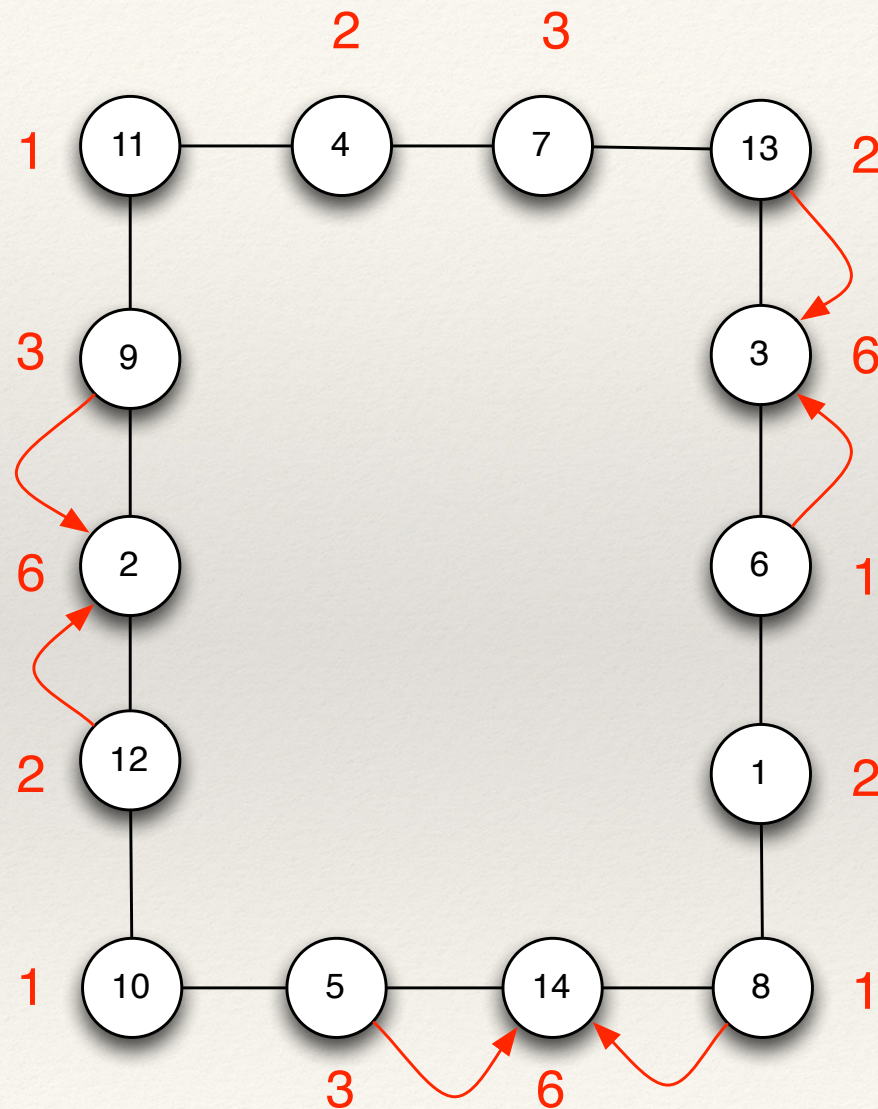
 envoyer $c(u)$ aux deux voisins de u

 recevoir $c(v)$ du voisin de gauche v et $c(w)$ du voisin de droite w

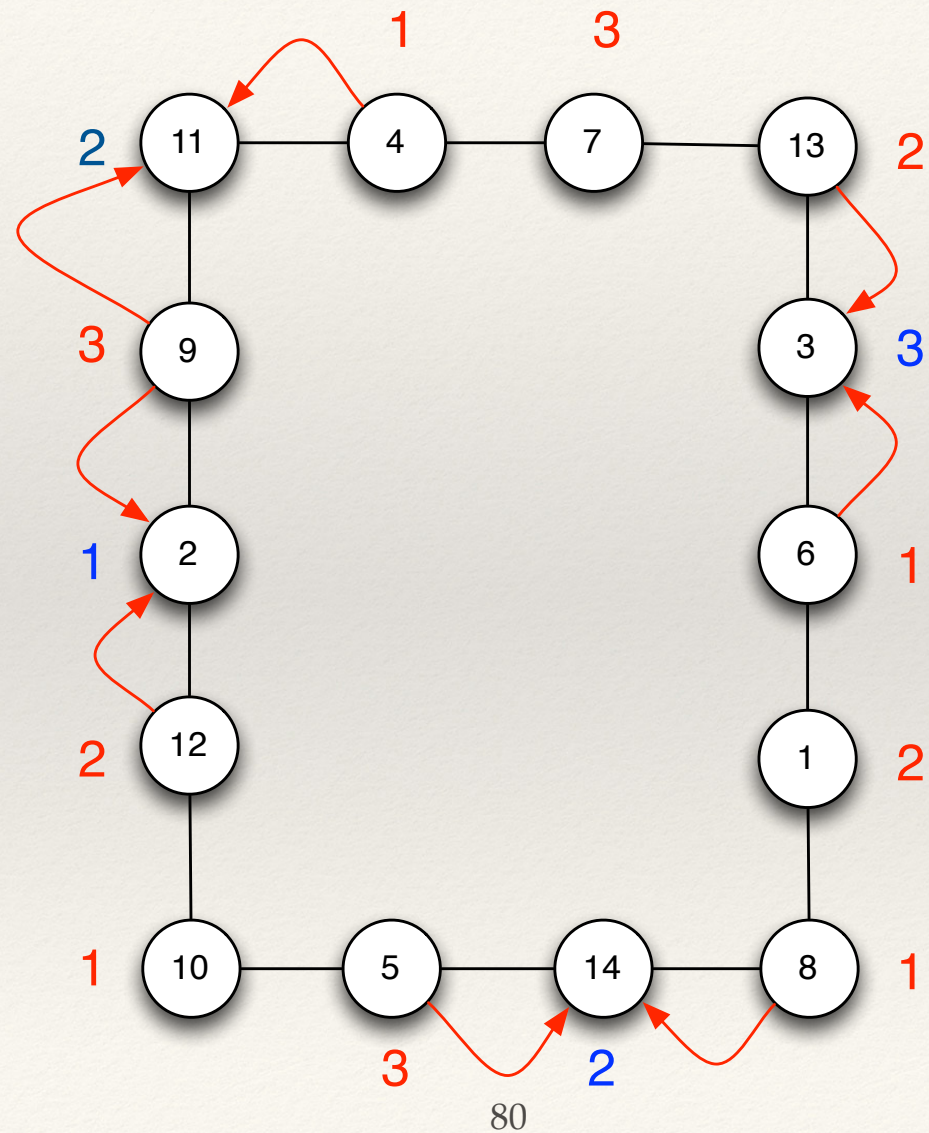
si $c(u) = i$ **alors** $c(u) \leftarrow \min\{c : c \in \{1, 2, 3\} \setminus \{c(v), c(w)\}\}$

fin.

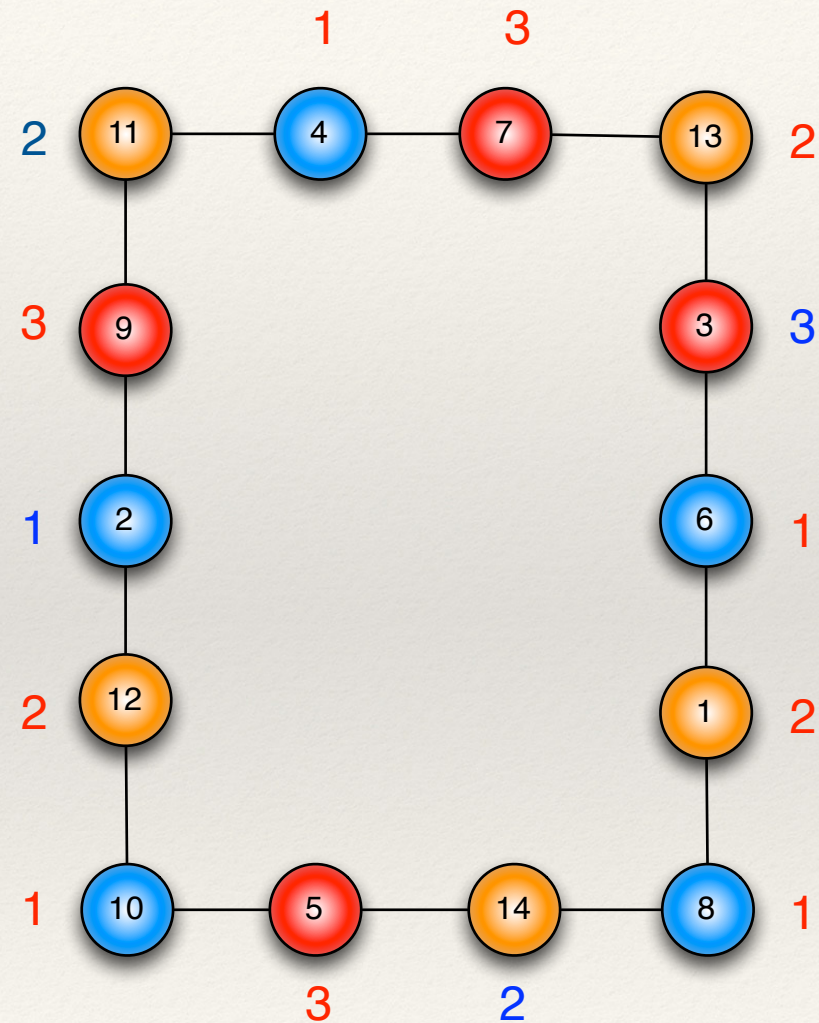
Exemple



Exemple



Exemple



Théorème

- ❖ Il existe un algorithme de $(\Delta+1)$ -coloration de l'anneau C_n s'exécutant en $1/2 \log^* n + O(1)$ étapes dans le modèle synchrone.
- ❖ Cet algorithme suppose que les noeuds ont une connaissance a priori de n , et ont une notion consistante de droite et de gauche.

Log-étoile

$$\forall x \leq 2: \log^* x := 1$$

$$\forall x > 2: \log^* x := 1 + \log^*(\log x)$$

Remarque

- ❖ Log-étoile est une fonction qui croît de façon «incroyablement» lentement.
- ❖ Log-étoile du nombre d'atomes observable dans l'univers (estimé à 10^{80}) est de 5.