Le problème **P** = **NP**

la complexité des algorithmes

Arnaud Durand

Université Paris 7

Fête de la science 2009

Problèmes du millénium

Depuis 2000, le Clay Institute propose un prix pour la résolution de 7 questions mathématiques.

- La conjecture de Birch et Swinnerton-Dyer
- La conjecture de Hodge
- Les équations de Navier-Stokes
- **P** vs **NP**
- La conjecture de Poincaré
- L'hypothèse de Riemann
- La théorie de Yang-Mills

P vs NP

- Question de mathématique et d'informatique.
- Question ouverte depuis environ 40 ans mais aux origines bien plus lointaines...

Questions

Qu'est-ce qu'une fonction mathématique calculable ? une propriété ou un problème décidable ?
Qu'est-ce qu'un algorithme ?

Propriété, problème : réponse = oui ou non

Algorithme

Origine : Al-Khawarizmi (mathématicien, géographe et astronome perse du 9ème siècle).
Sens : recette, méthode pour résoudre un problème ou calculer une fonction.

Algorithme

Origine : Al-Khawarizmi (mathématicien, géographe et astronome perse du 9ème siècle).
Sens : recette, méthode pour résoudre un problème ou calculer une fonction.

Exemple : la multiplication d'entiers

$$\begin{array}{rrrr} & 1 & 2 & 3 \\ \times & & 1 & 5 \\ \hline \end{array}$$

Algorithme

Origine : Al-Khawarizmi (mathématicien, géographe et astronome perse du 9ème siècle).
Sens : recette, méthode pour résoudre un problème ou calculer une fonction.

Exemple : la multiplication d'entiers

$$\begin{array}{cccc} & & 1 & 1 & \\ & & 1 & 2 & 3 \\ \times & & & 1 & 5 \\ \hline & & 6 & 1 & 5 \end{array}$$

Algorithme

Origine : Al-Khawarizmi (mathématicien, géographe et astronome perse du 9ème siècle).
Sens : recette, méthode pour résoudre un problème ou calculer une fonction.

Exemple : la multiplication d'entiers

$$
\begin{array}{cccc}
 & & {\scriptstyle 1}\ {\scriptstyle 1} & \\
 & & 1\ 2\ 3 & \\
\times & & 1\ 5 & \\
\hline
 & & 6\ 1\ 5 & \\
 & 1\ 2\ 3 & & \\
\hline
\end{array}
$$

Algorithme

Origine : Al-Khawarizmi (mathématicien, géographe et astronome perse du 9ème siècle).
Sens : recette, méthode pour résoudre un problème ou calculer une fonction.

Exemple : la multiplication d'entiers

$$\begin{array}{cccc} & & 1 & 1 & \\ & & 1 & 2 & 3 \\ \times & & & 1 & 5 \\ \hline & & 6 & 1 & 5 \\ & 1 & 2 & 3 & \\ \hline & 1 & 8 & 4 & 5 \end{array}$$

Algorithme

Origine : Al-Khawarizmi (mathématicien, géographe et astronome perse du 9ème siècle).
Sens : recette, méthode pour résoudre un problème ou calculer une fonction.

Exemple : la multiplication d'entiers

$$\begin{array}{cccc} & & 1 & 1 & \\ & & 1 & 2 & 3 \\ \times & & & 1 & 5 \\ \hline & & 6 & 1 & 5 \\ & 1 & 2 & 3 & \\ \hline & 1 & 8 & 4 & 5 \end{array}$$

Besoin de :

- actions répétitives (itérations)
- actions conditionnelles (tests)
- résultats intermédiaires

La longueur du calcul dépend de la taille des entiers.

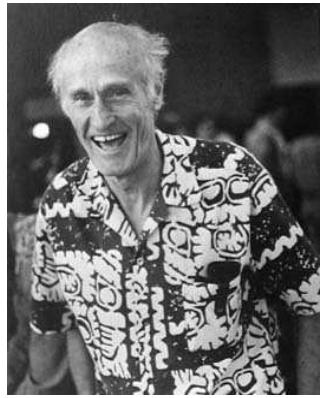
Modèles de calcul et fonction calculable

Début des années 1930 :

- formalisation de la notion de fonction calculable et d'algorithme. Approches équivalentes (machines de Turing, fonctions récursives, lambda calcul) et actuelles...

Alan Turing

Alonzo Church



Stephen C. Kleene

- mise en évidence des limites : il existe des fonctions non calculables (et des problèmes indécidables) !

Problèmes indécidables

Le problème suivant est indécidable :

Entrée : un programme P, une entrée x

Question : l'exécution de P sur l'entrée x s'arrête-elle ?

Problème assez naturel !

Un des premiers d'une longue lignée dans de nombreux domaines.

Algorithme et complexité

Petit à petit, le débat se déplace. La question :
“Qu’est-ce qu’une fonction calculable ?”
devient, par exemple :
“Si une fonction est calculable, peut-on mesurer le “coût” (sa complexité) nécessaire pour la calculer ?”

Coût / complexité :

nombre d’étapes, d’opérations élémentaires, de l’algorithme (”temps abstrait“).
Rabin, Cobham, Blum, Edmonds, Hartmanis, Stearns (milieu des années 60)

Multiplication : $m \times n$ ”petites” multiplications et additions pour des entiers de tailles m et n.

Algorithme et complexité

Regrouper les problèmes par catégorie....

Si je sais calculer un ensemble de fonctions avec telle ou telle ressource, puis-je le faire aussi avec telle autre ressource (moins d'étapes de calcul, avec telle mémoire, avec un autre type de machine) ?

P vs *NP*

Le problème **P** vs **NP** porte sur deux classes de problèmes :

- **P** : les problèmes “faciles” à décider, c’est à dire de complexité raisonnable.
- **NP** : les problèmes “faciles” à vérifier.

Tout cela reste à définir...

Problèmes

Entrée : trois entiers n, p, q

Question : $n = p \times q$?

Entrée : un entier n

Question : n est-il premier ?

Entrée : un entier n

Question : existe-t-il des entiers non nuls x, y et z tels que $x^n + y^n = z^n$?

Problèmes

Entrée : une liste de mots $l_1, l_2, \ldots, l_n$

Sortie : renvoyer la liste triée par ordre croissant ?

Entrée : un programme P, une entrée x

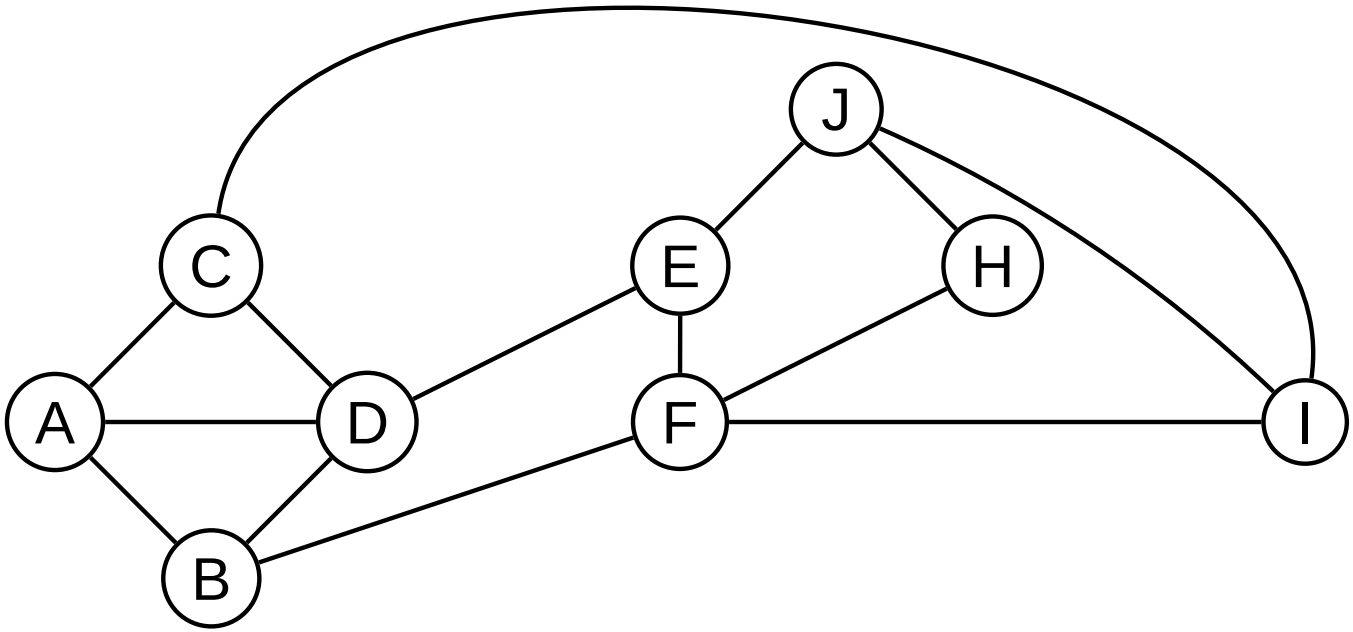
Question : l'execution de P sur l'entrée x s'arrête-elle ?

Un exemple : le circuit hamiltonien

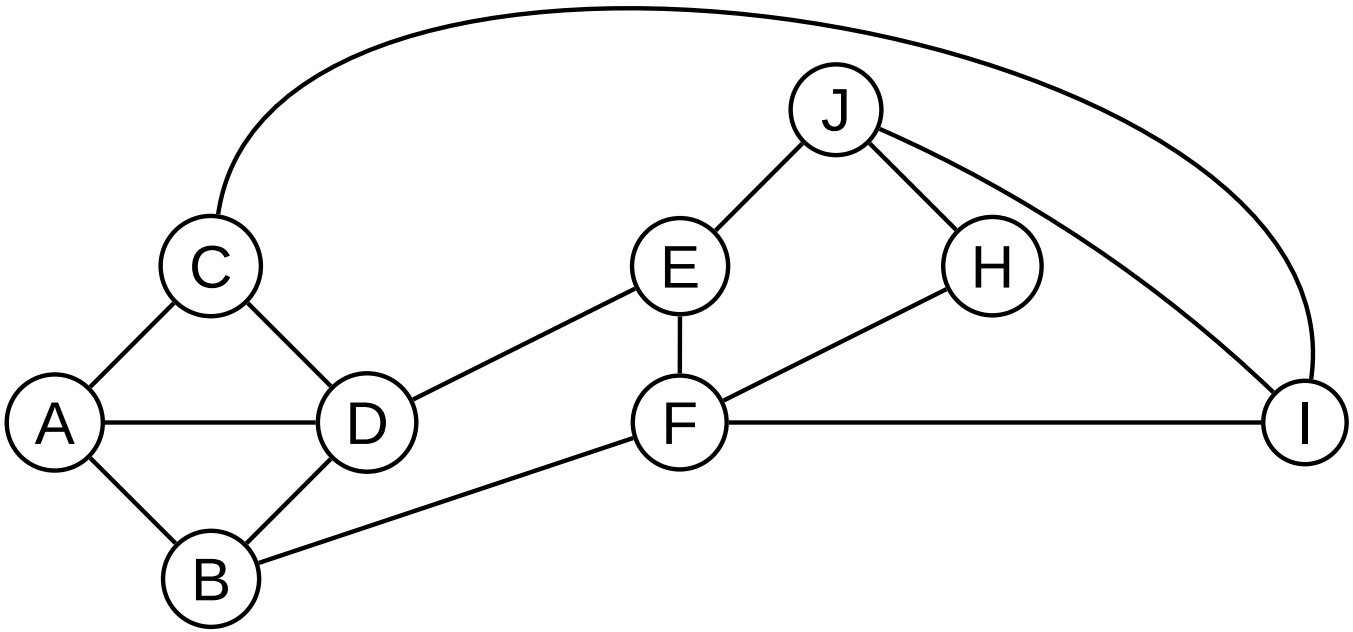
Circuit hamiltonien

Étant donné un graphe $G = (V, E)$ et un sommet $v \in V$, peut-on trouver un circuit empruntant des arêtes du graphe, commençant et finissant en v et passant une seule fois par tout autre sommet de G ?

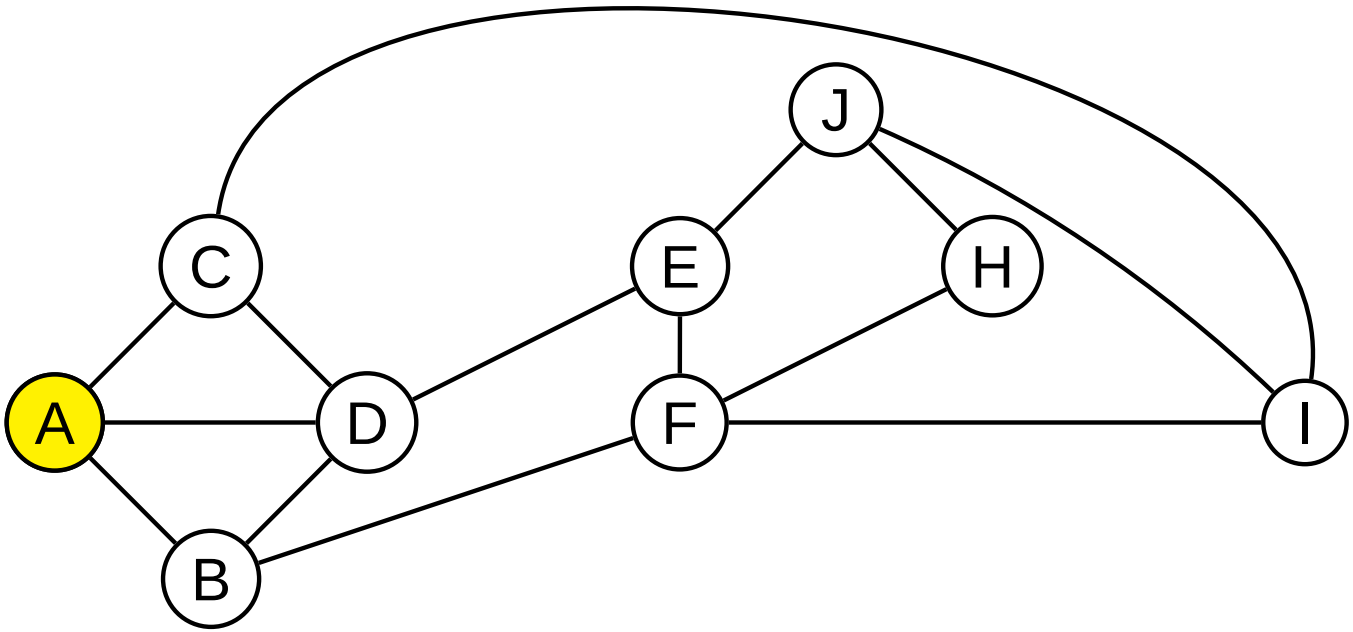
Aussi : problème des bandits, voyageur de commerce.



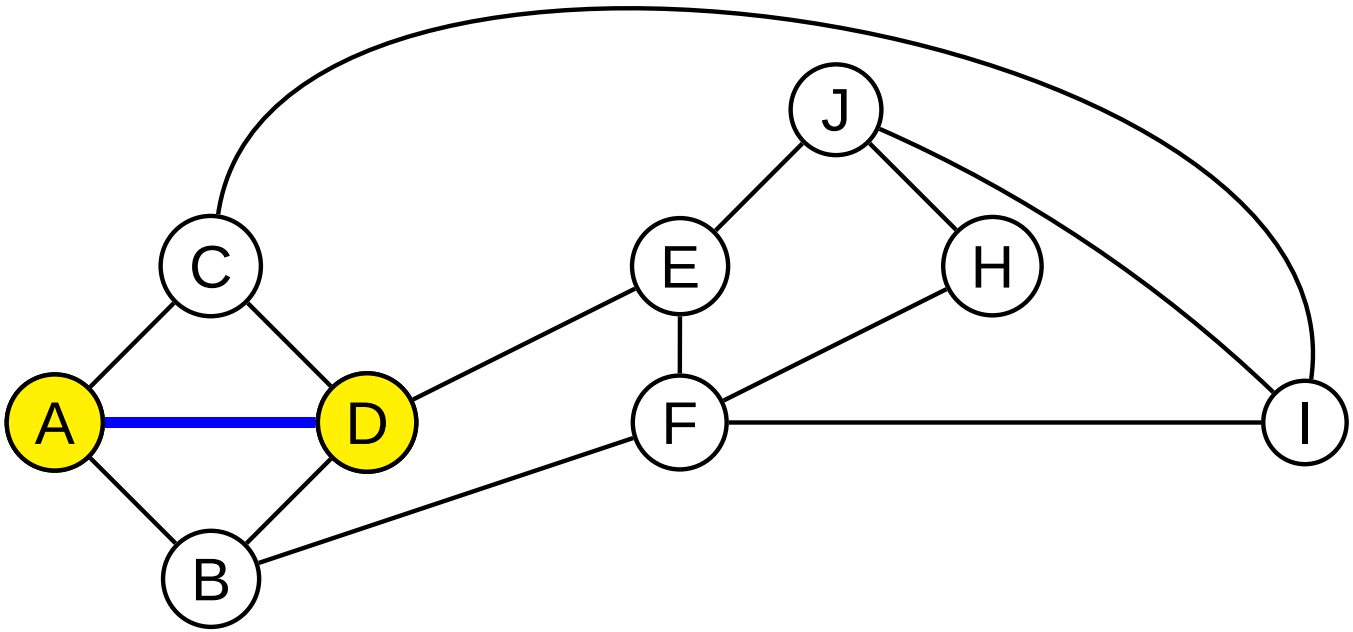
Un exemple : le circuit hamiltonien



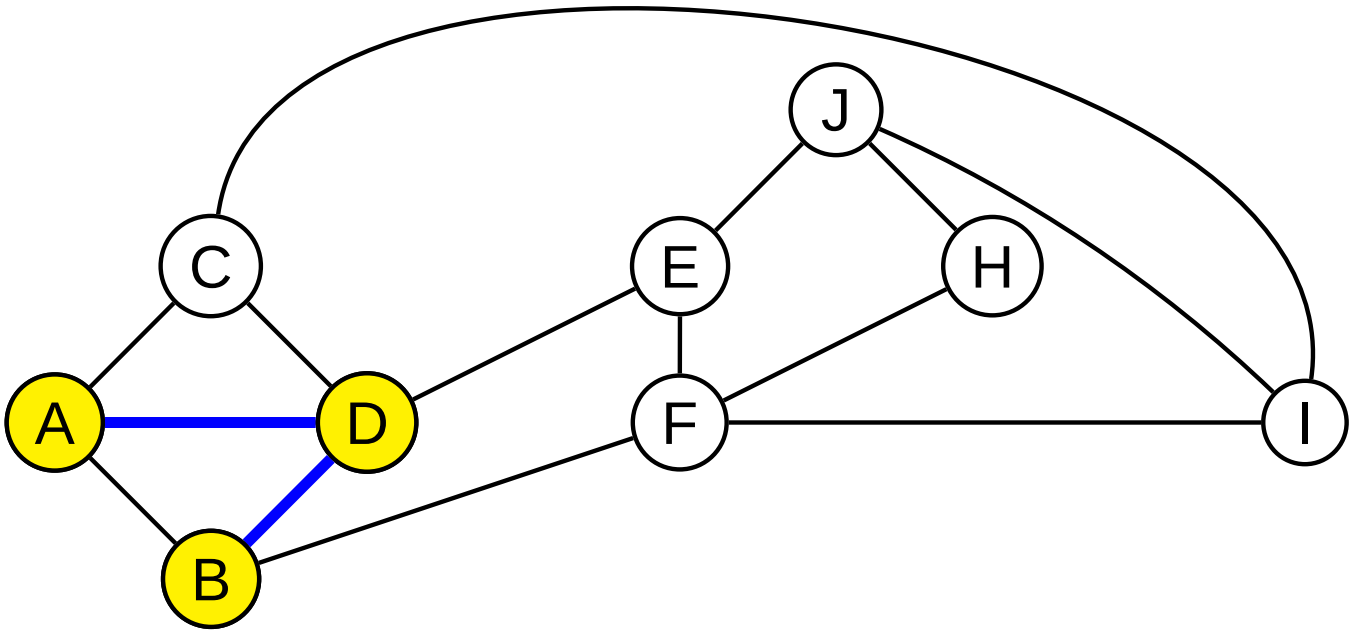
Un exemple : le circuit hamiltonien



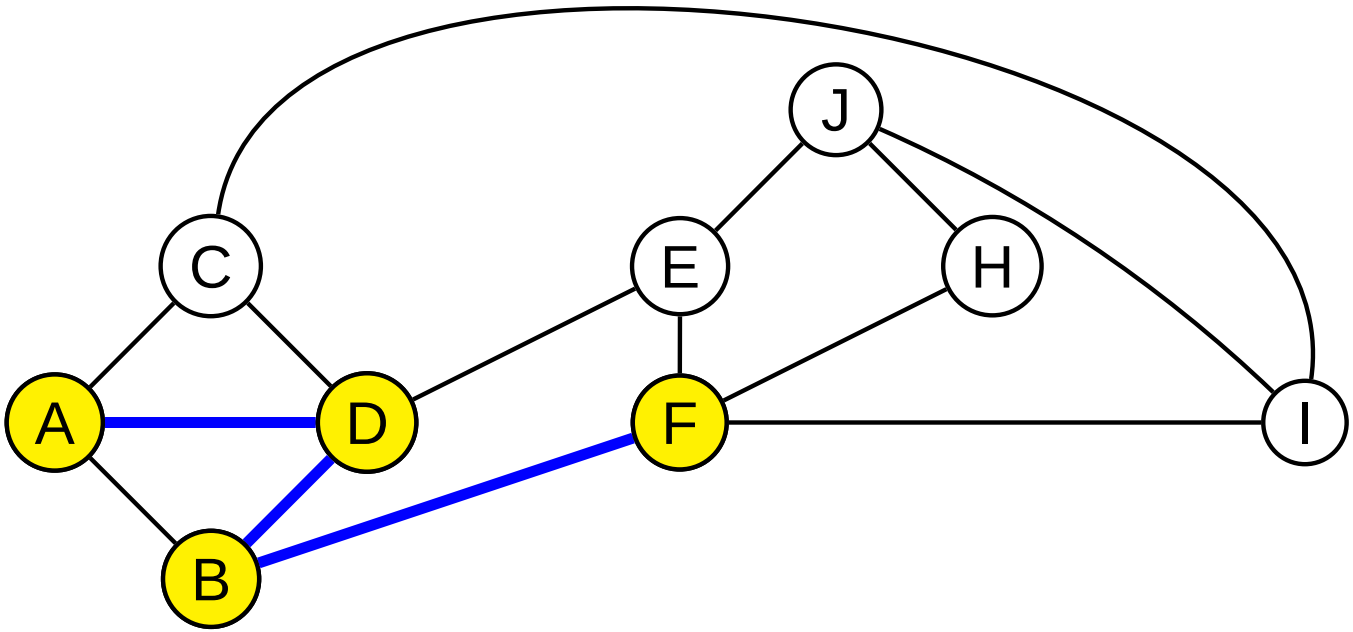
Un exemple : le circuit hamiltonien



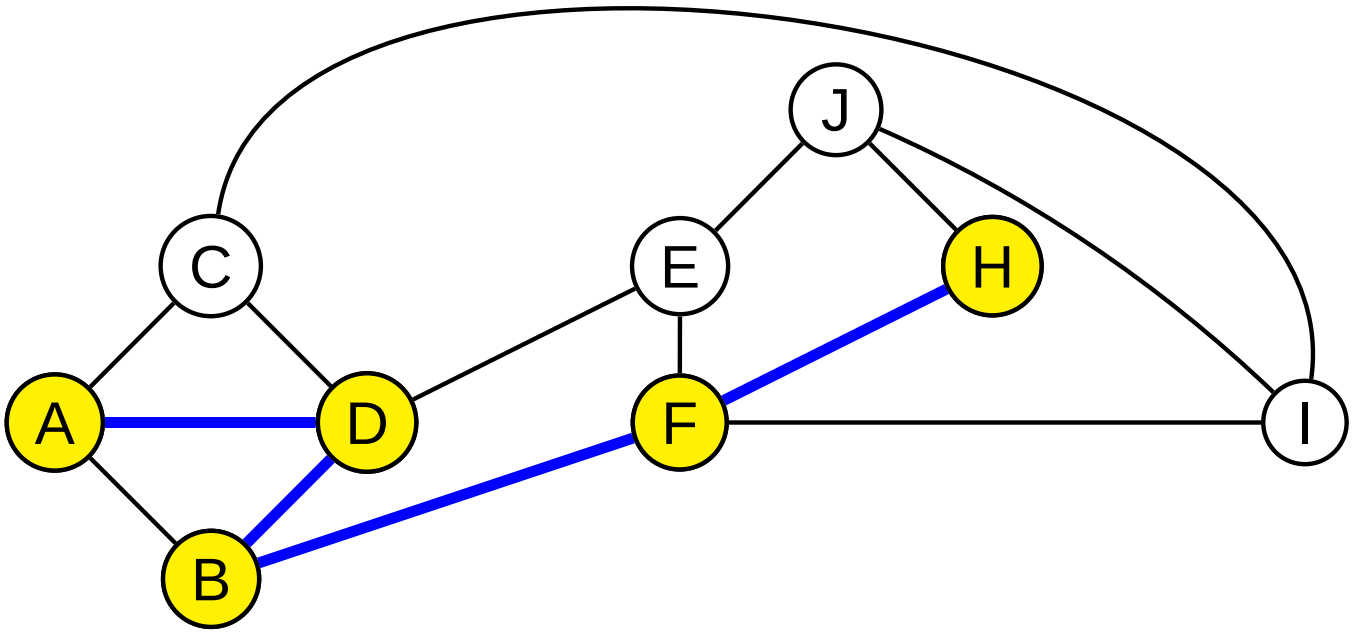
Un exemple : le circuit hamiltonien



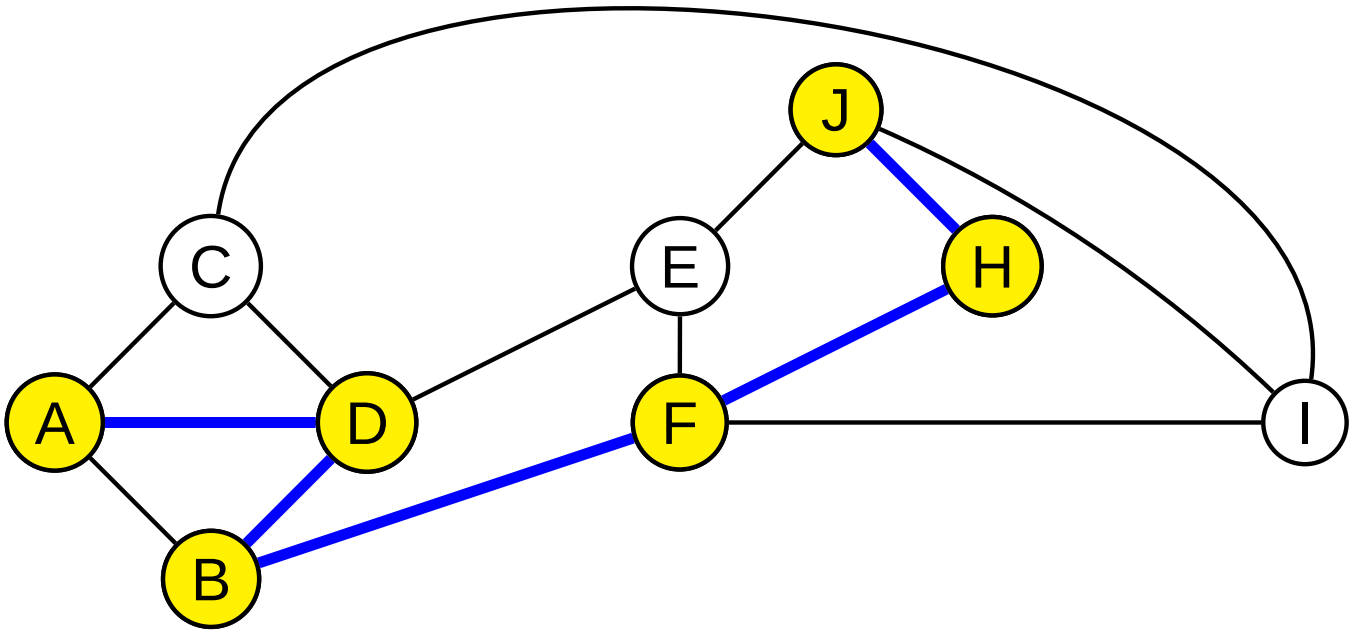
Un exemple : le circuit hamiltonien



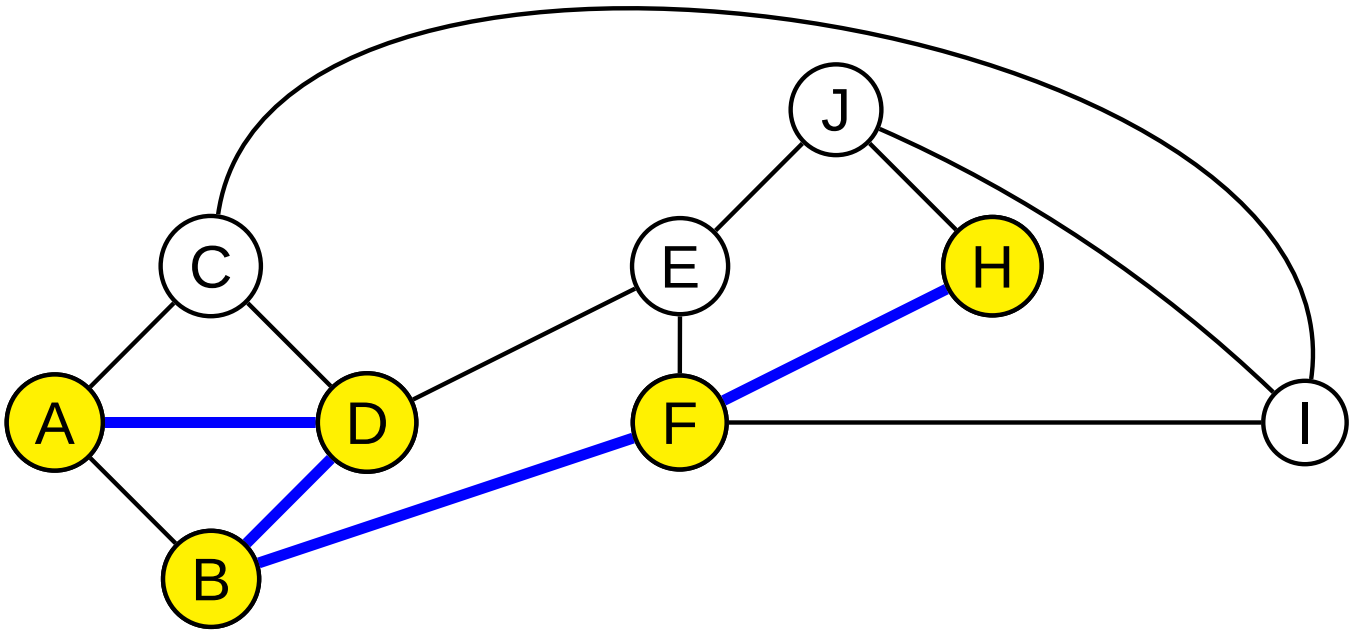
Un exemple : le circuit hamiltonien



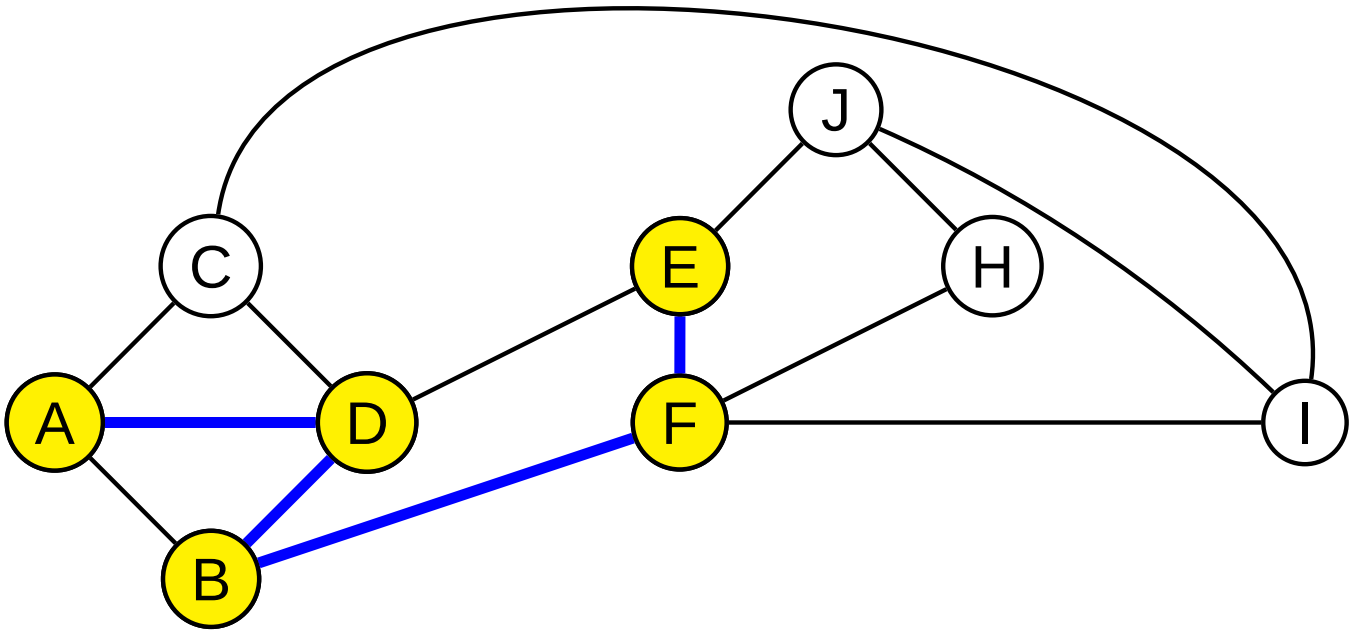
Un exemple : le circuit hamiltonien



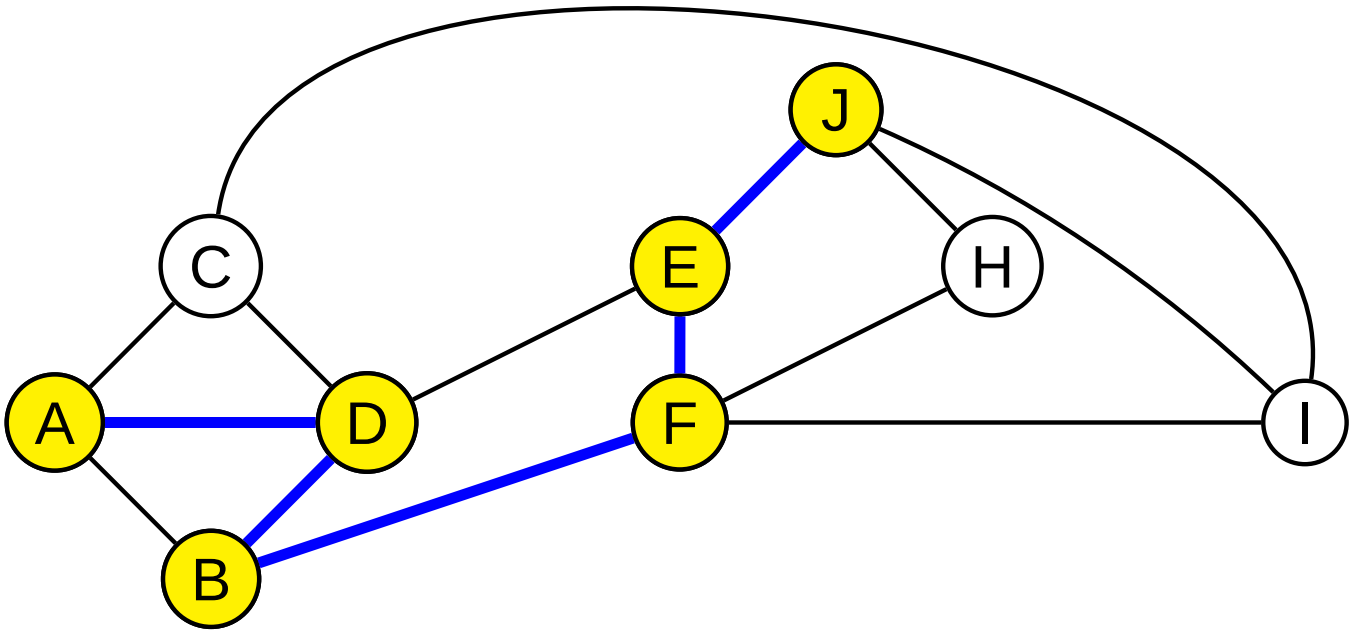
Un exemple : le circuit hamiltonien



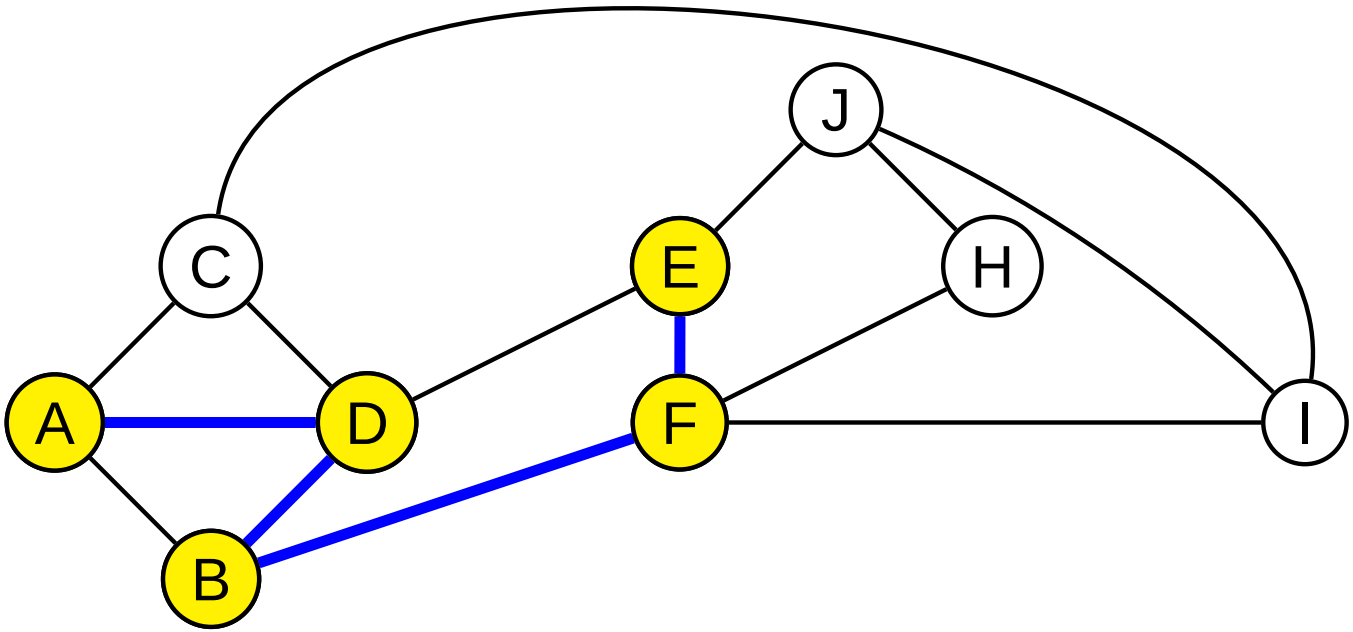
Un exemple : le circuit hamiltonien



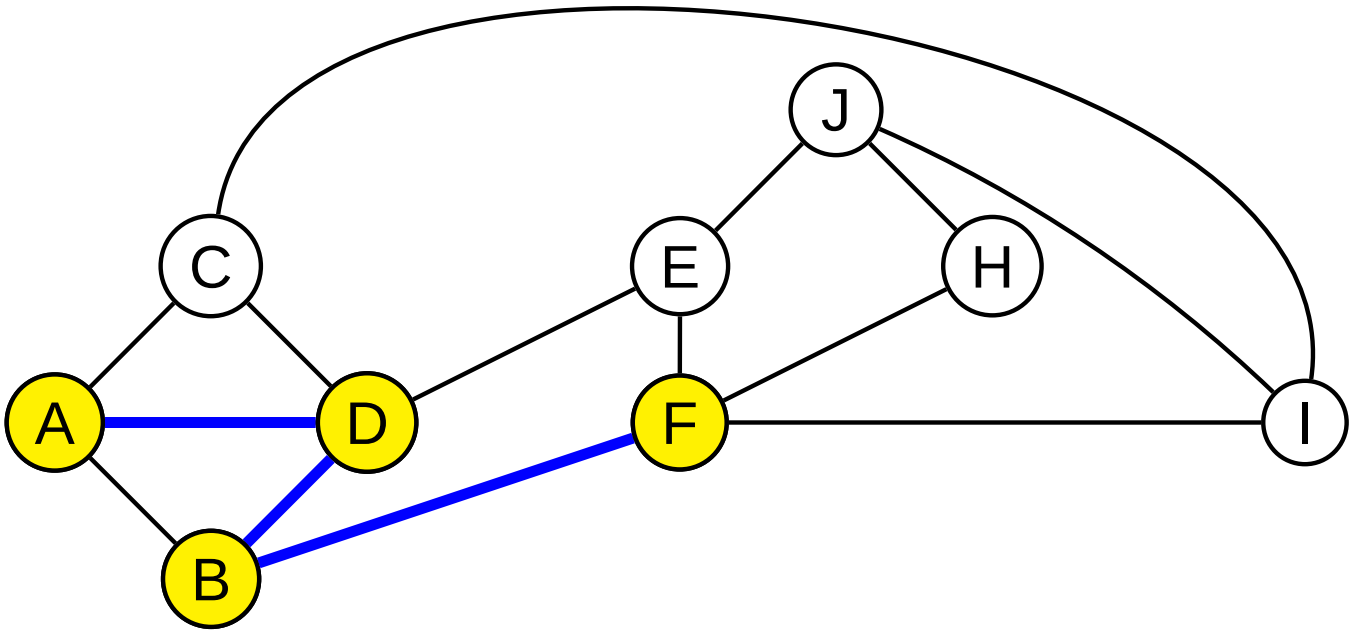
Un exemple : le circuit hamiltonien



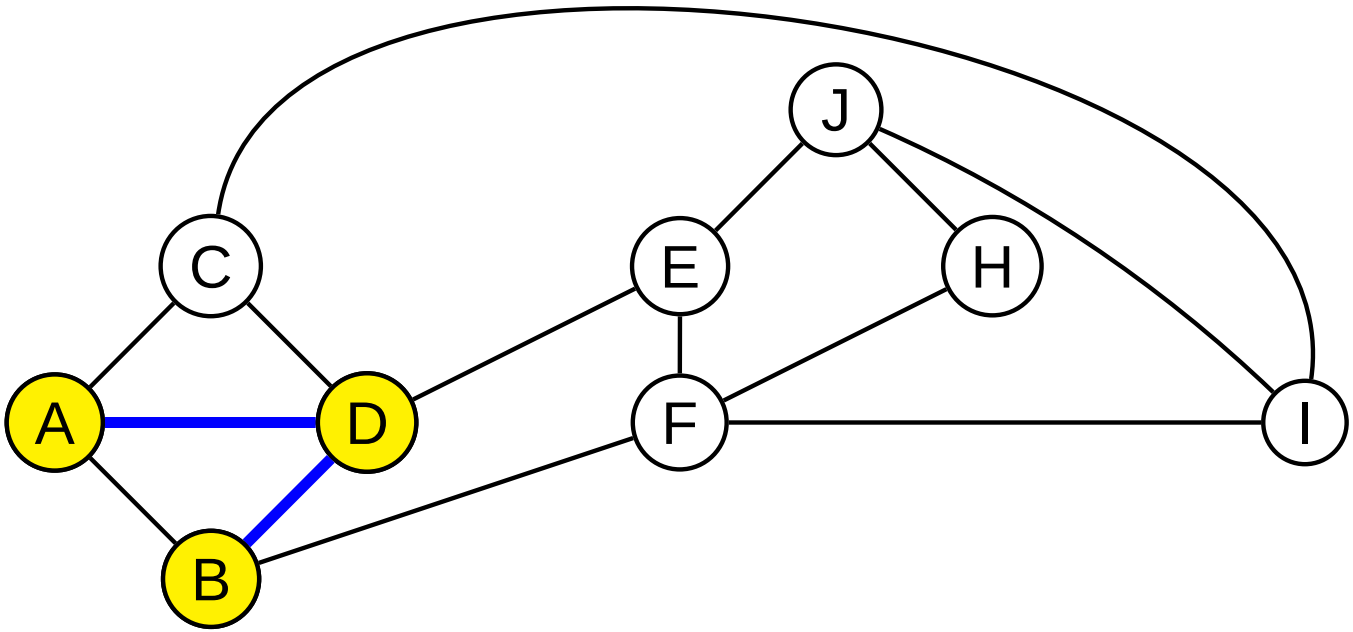
Un exemple : le circuit hamiltonien



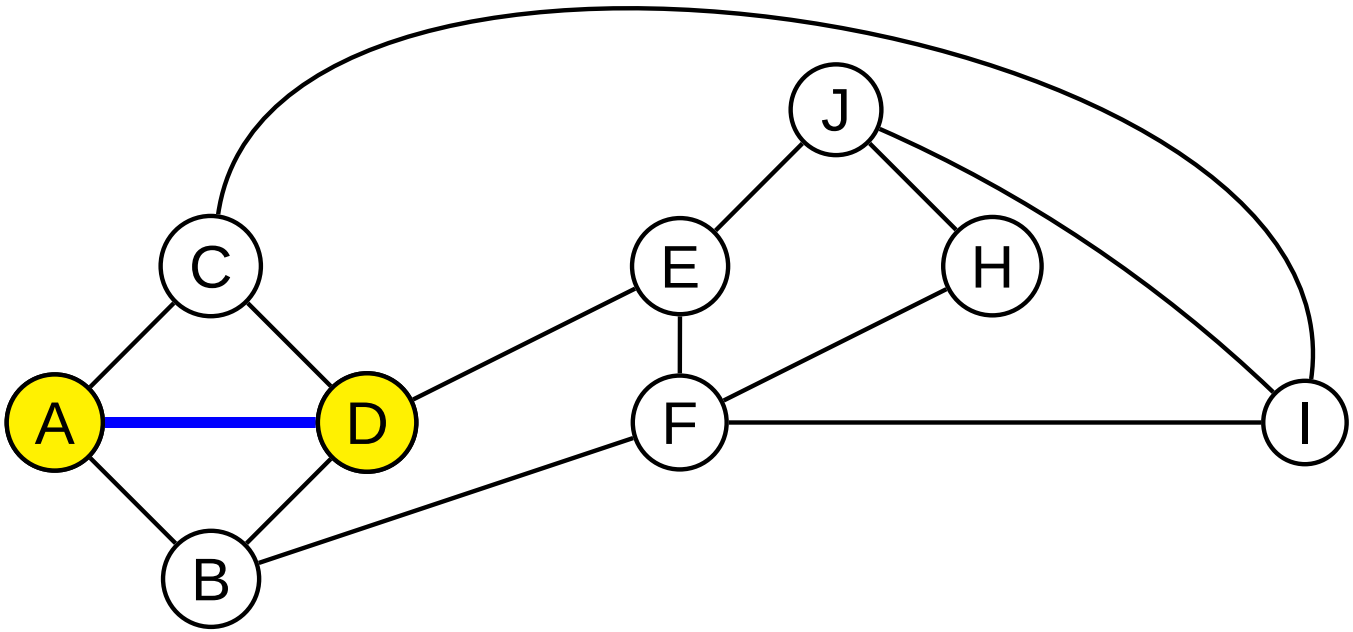
Un exemple : le circuit hamiltonien



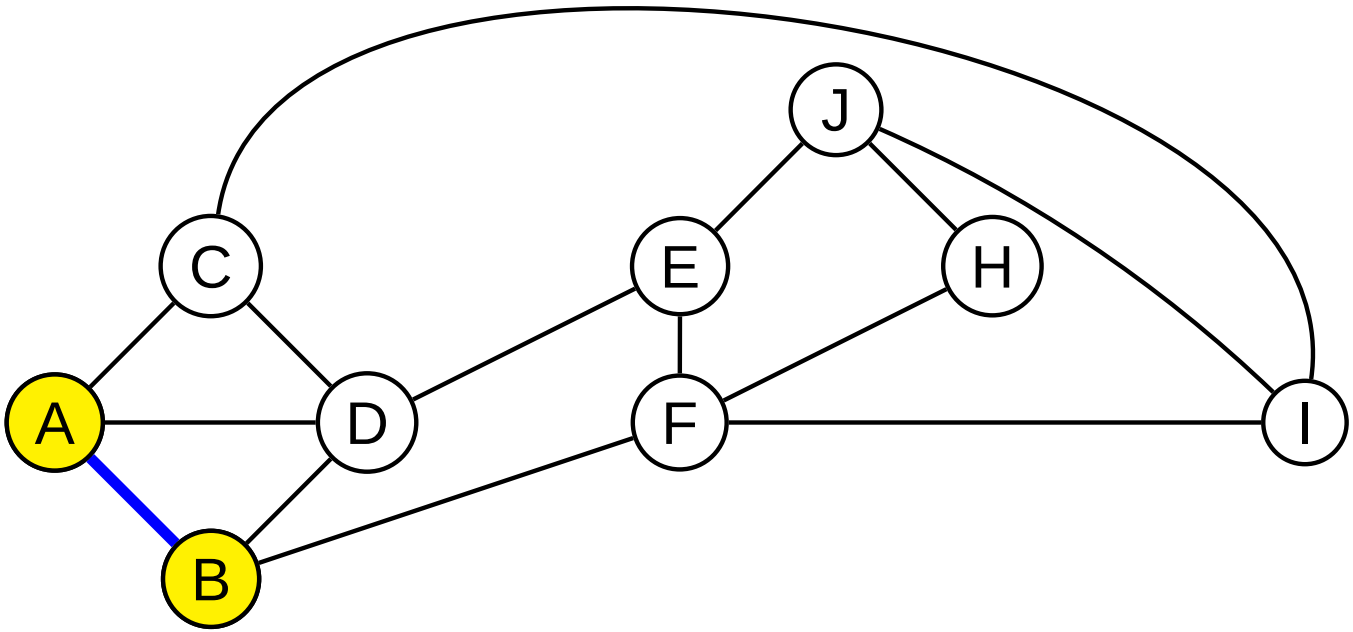
Un exemple : le circuit hamiltonien



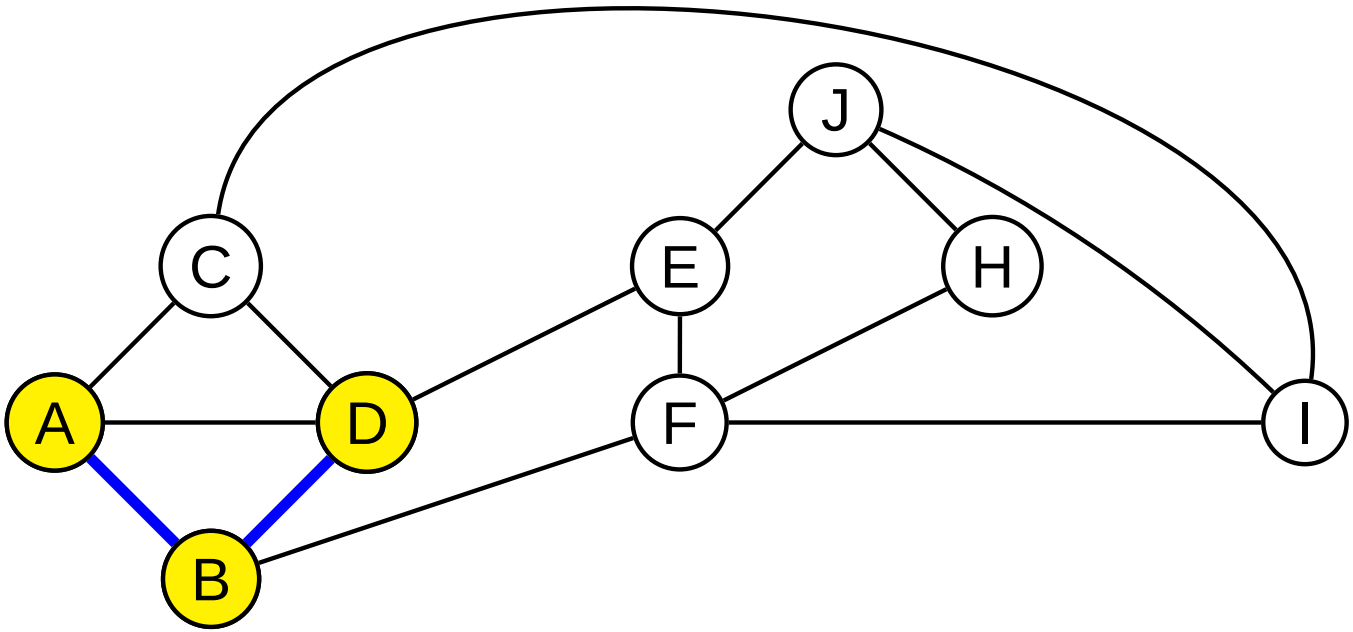
Un exemple : le circuit hamiltonien



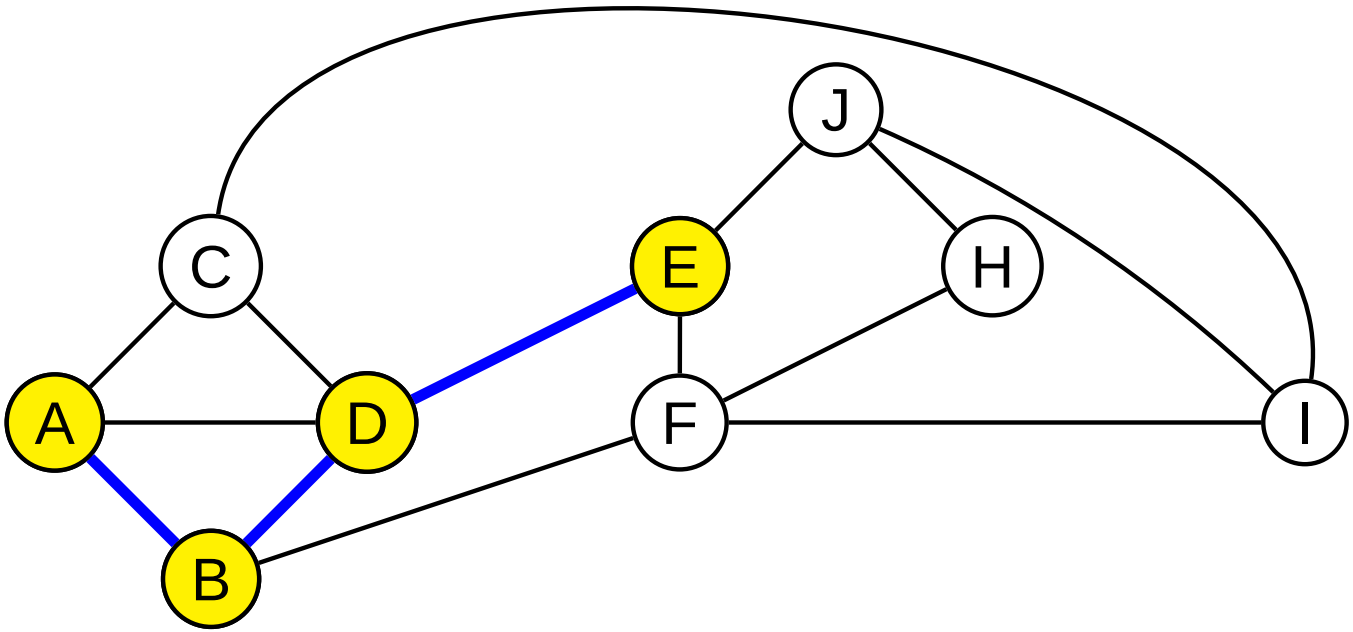
Un exemple : le circuit hamiltonien



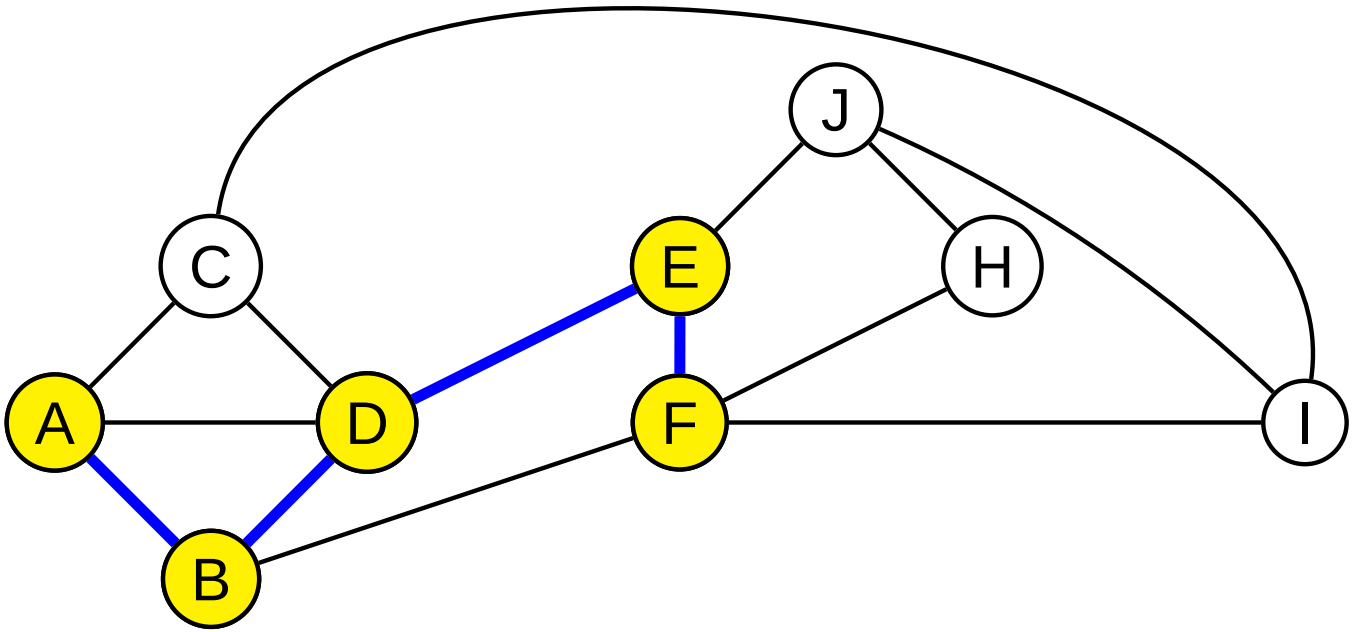
Un exemple : le circuit hamiltonien



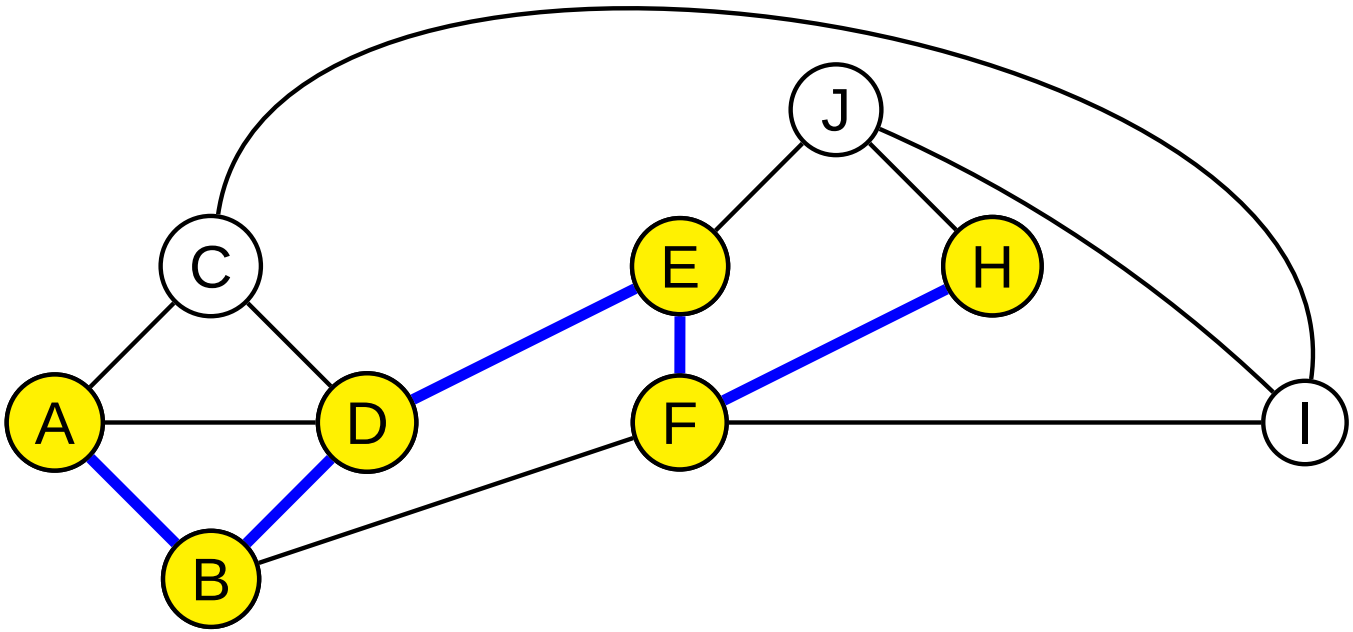
Un exemple : le circuit hamiltonien



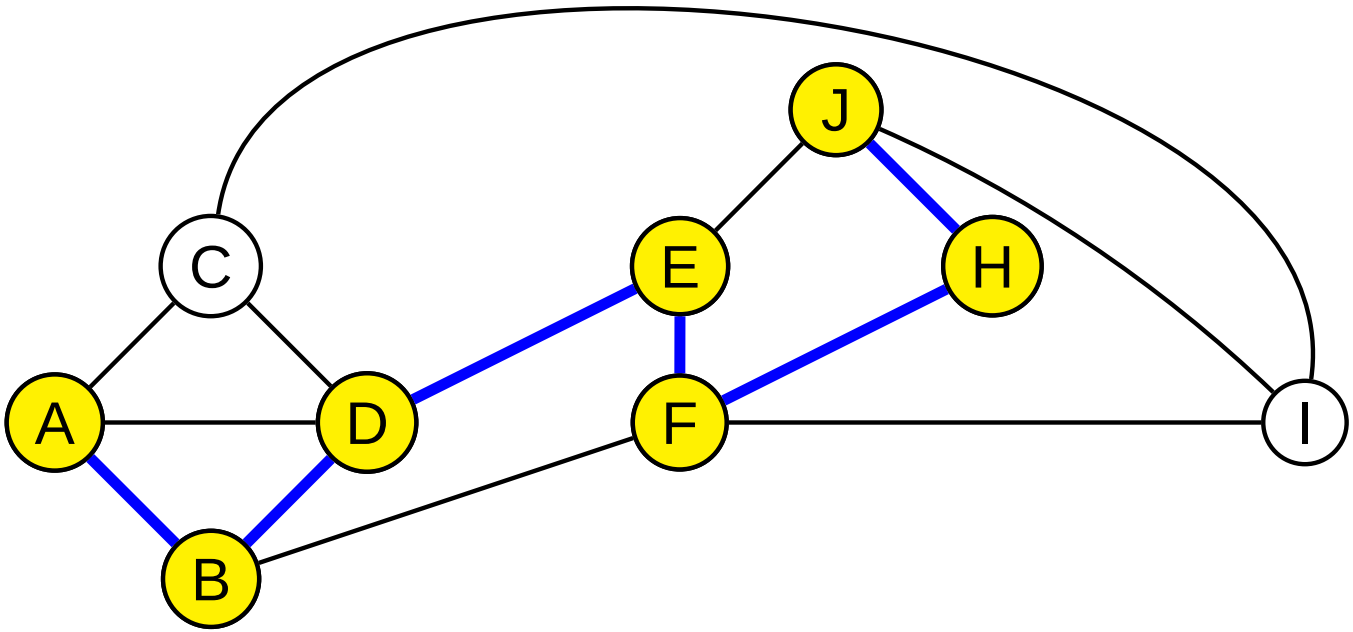
Un exemple : le circuit hamiltonien



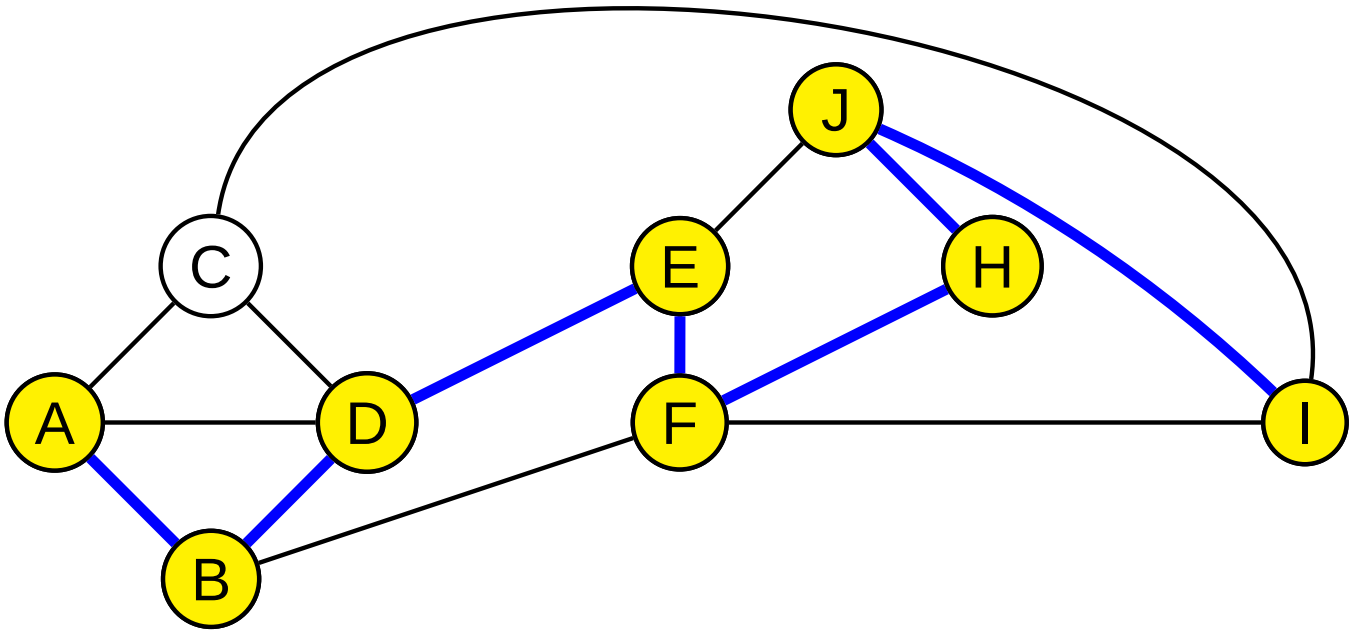
Un exemple : le circuit hamiltonien



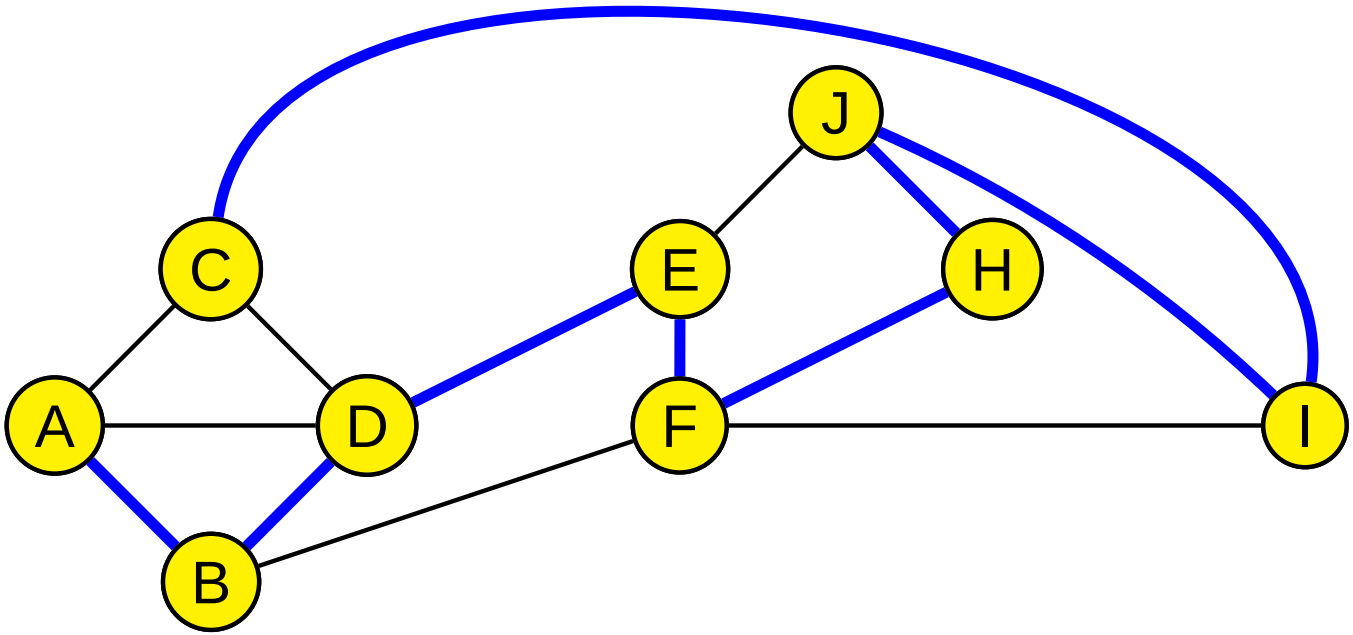
Un exemple : le circuit hamiltonien



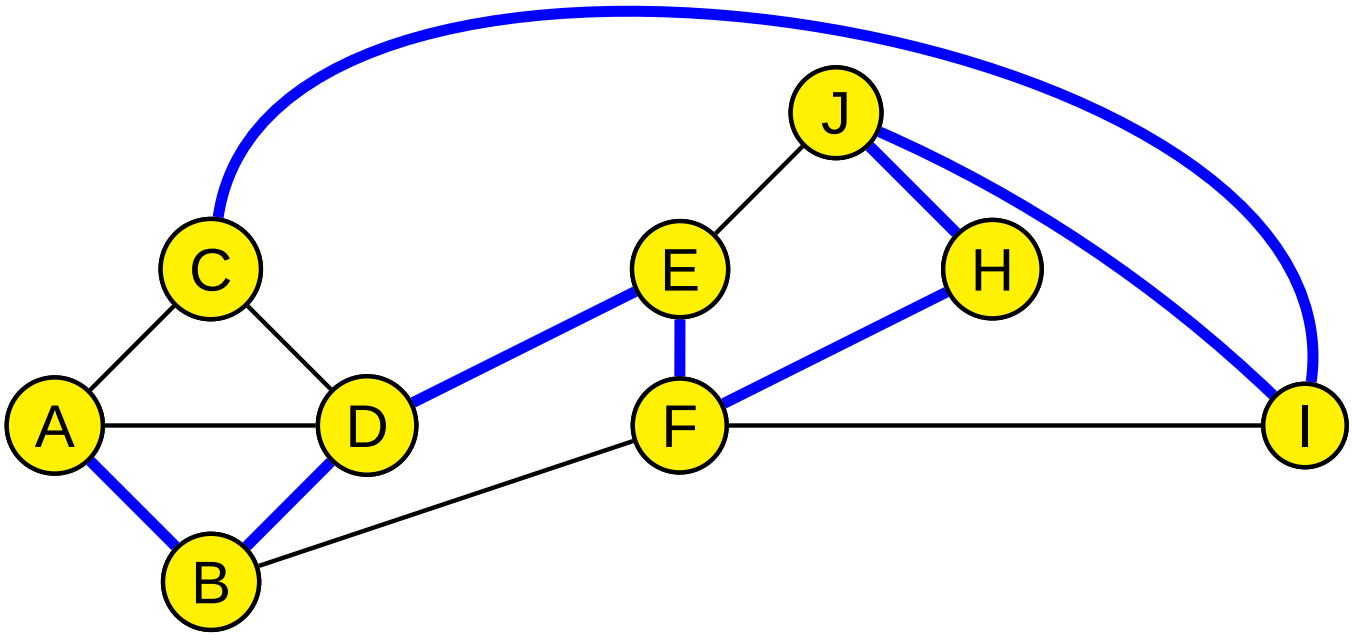
Un exemple : le circuit hamiltonien



Un exemple : le circuit hamiltonien



Un exemple : le circuit hamiltonien



Circuit hamiltonien : bilan

Problème difficile !

Taille du graphe $=$ nombre d'arêtes $+$ nombre de sommets $= |E| + |V|$

Nombre d'étapes

On peut être amené à essayer tous les chemins possibles... Il y en a :

$$(n-1)(n-2)\cdots 2 = (n-1)! \text{ avec } n = |V|$$

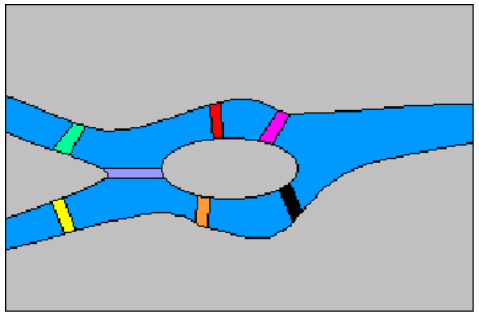
Pas de stratégie franchement meilleure (en dessous de 2^n) connue...

Le nombre d'étapes est exponentiel en la taille du graphe.

Plus facile : le circuit eulerien

Circuit eulerien

Étant donné un graphe $G = (V, E)$, peut-on trouver un circuit passant une seule fois par chaque arête du graphe ?

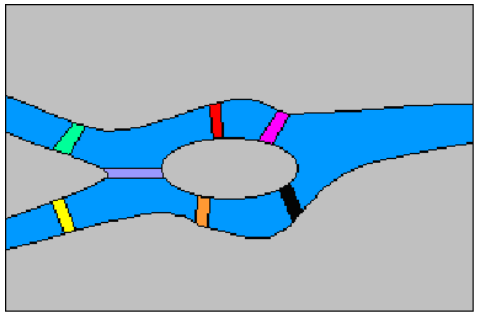


Les 7 ponts de Koenigsberg

Plus facile : le circuit eulerien

Circuit eulerien

Étant donné un graphe $G = (V, E)$, peut-on trouver un circuit passant une seule fois par chaque arête du graphe ?



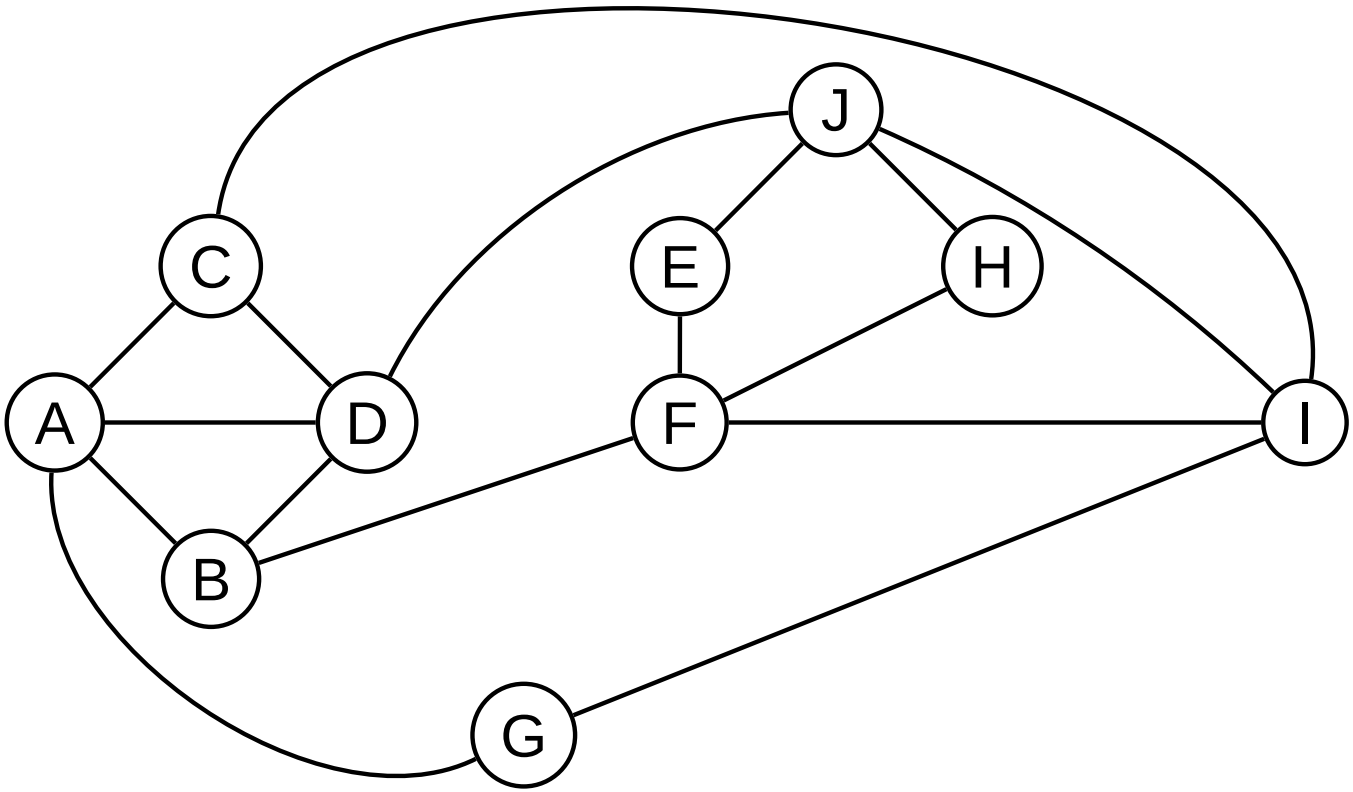
Les 7 ponts de Koenigsberg

Caractérisation

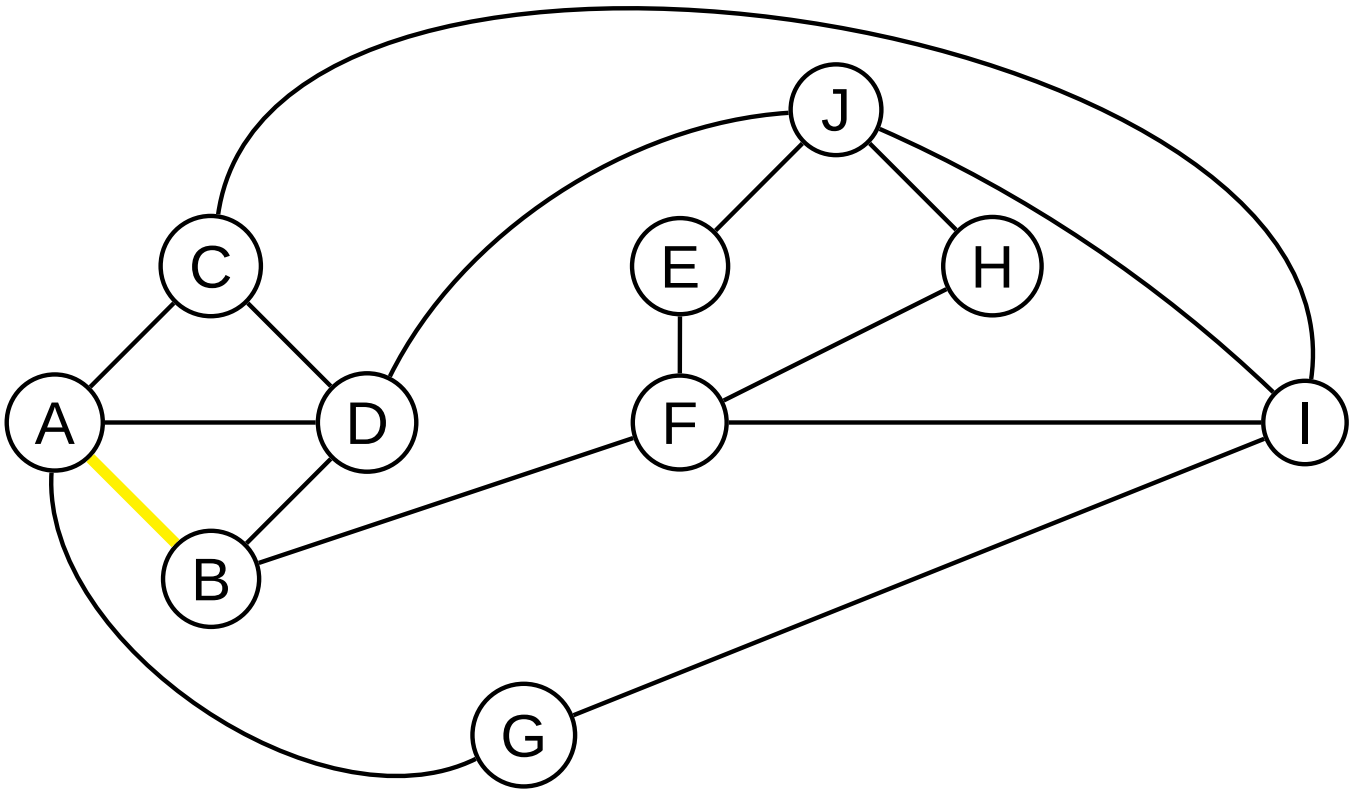
Un graphe $G = (V, E)$ admet un circuit eulerien si et seulement si tout sommet $v \in V$ est de degré pair.

degré d'un sommet v : nombre de sommets reliés à v (ses "voisins")

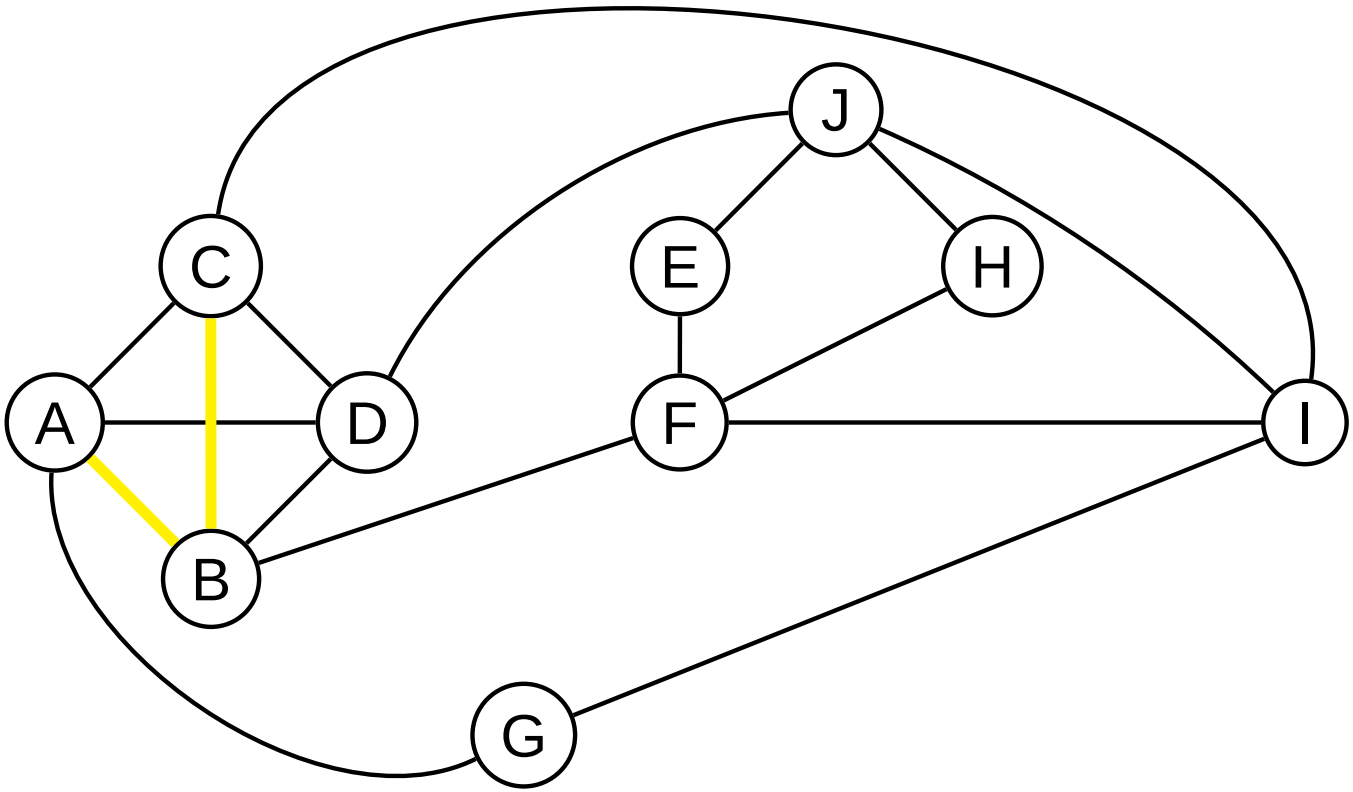
Circuit eulerien



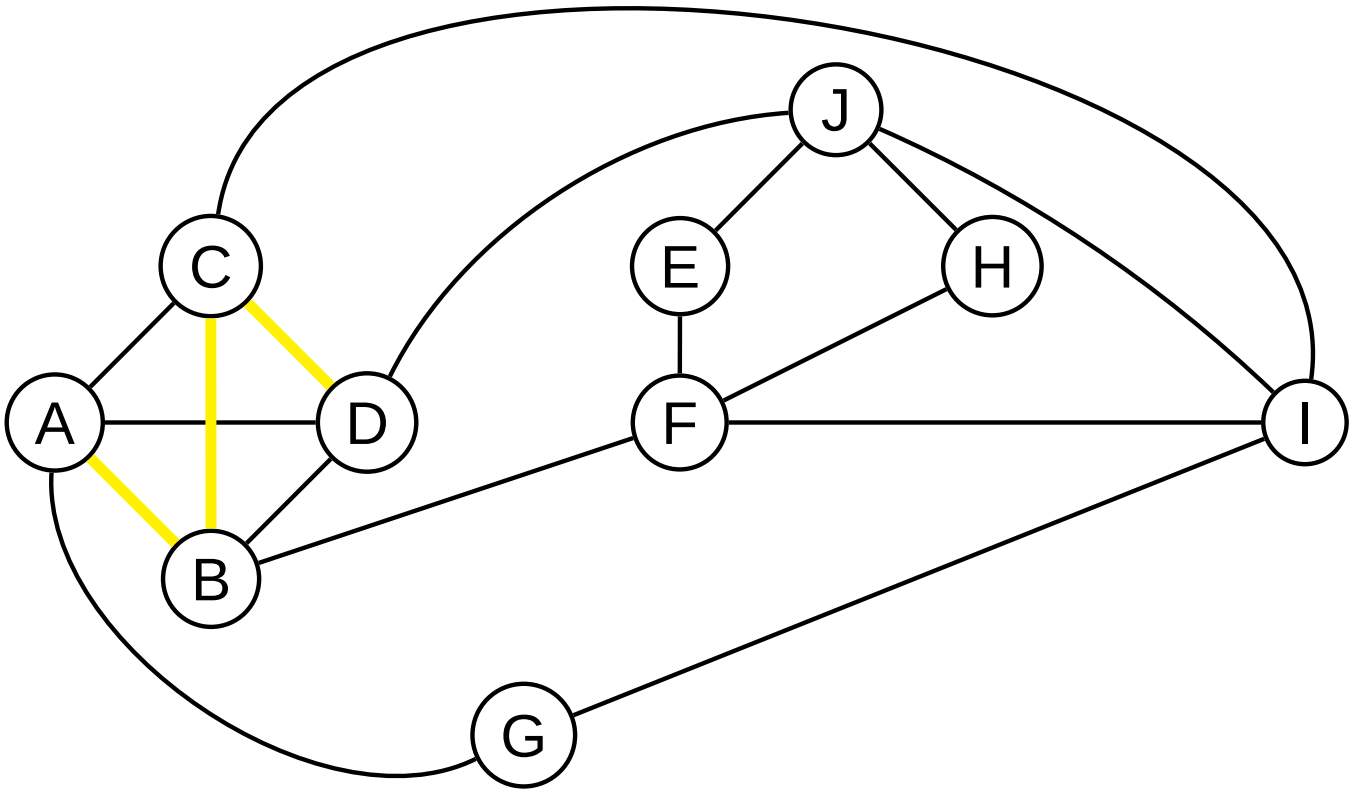
Circuit eulerien



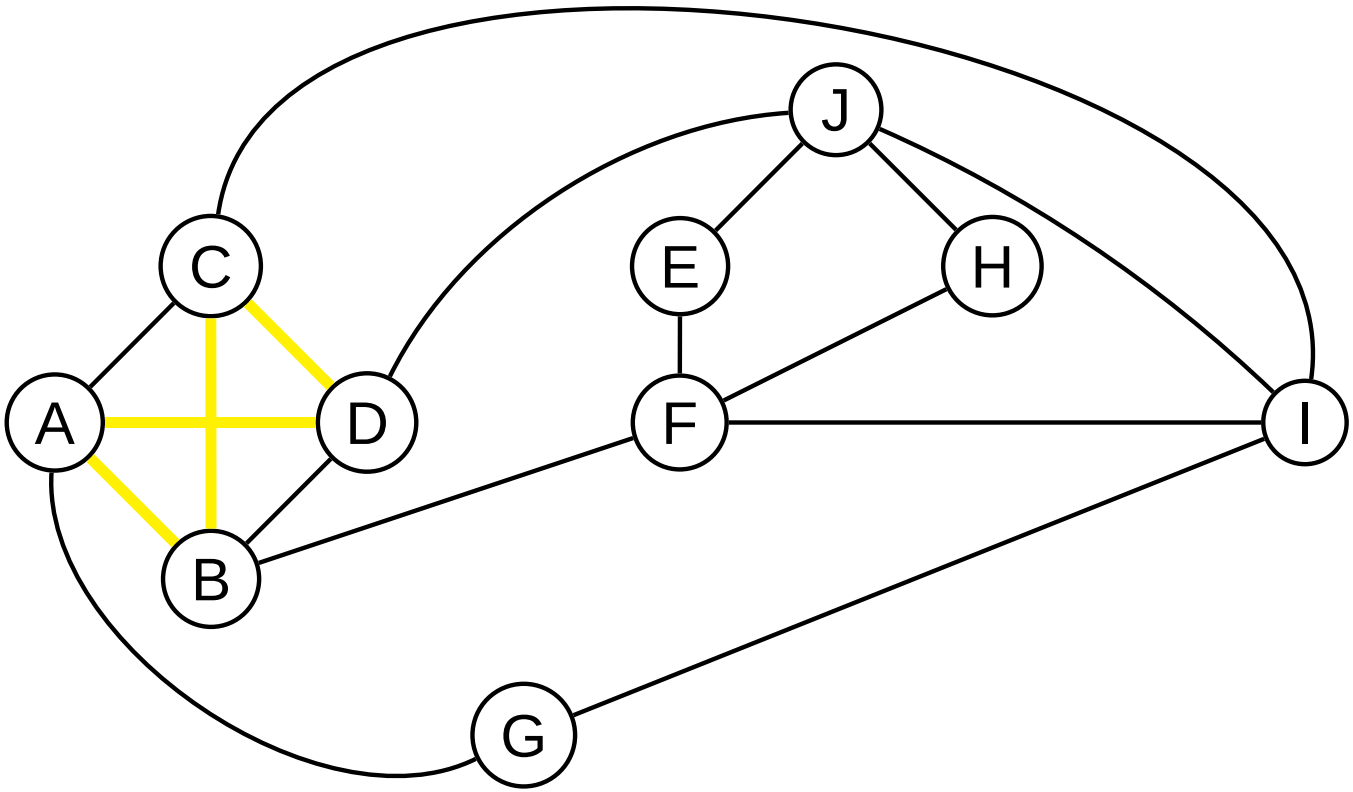
Circuit eulerien



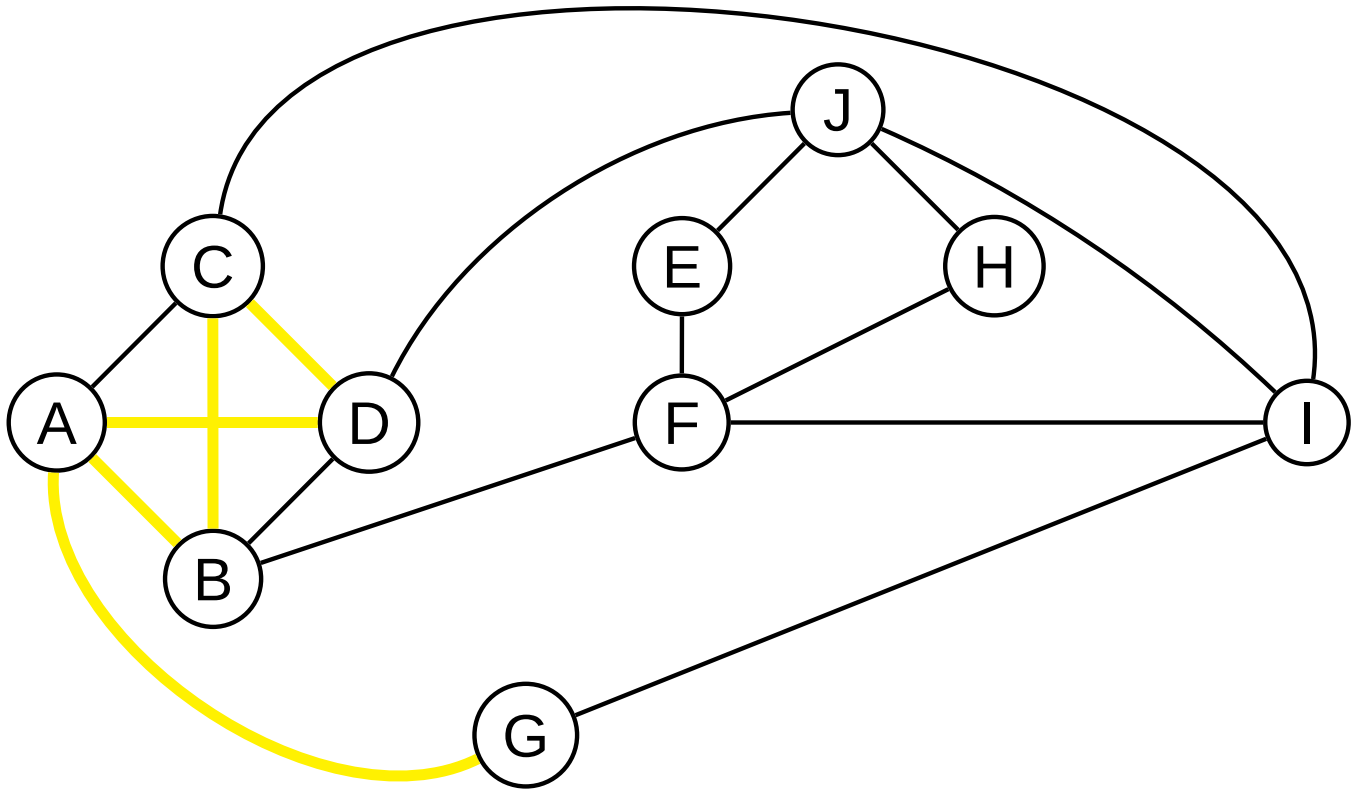
Circuit eulerien



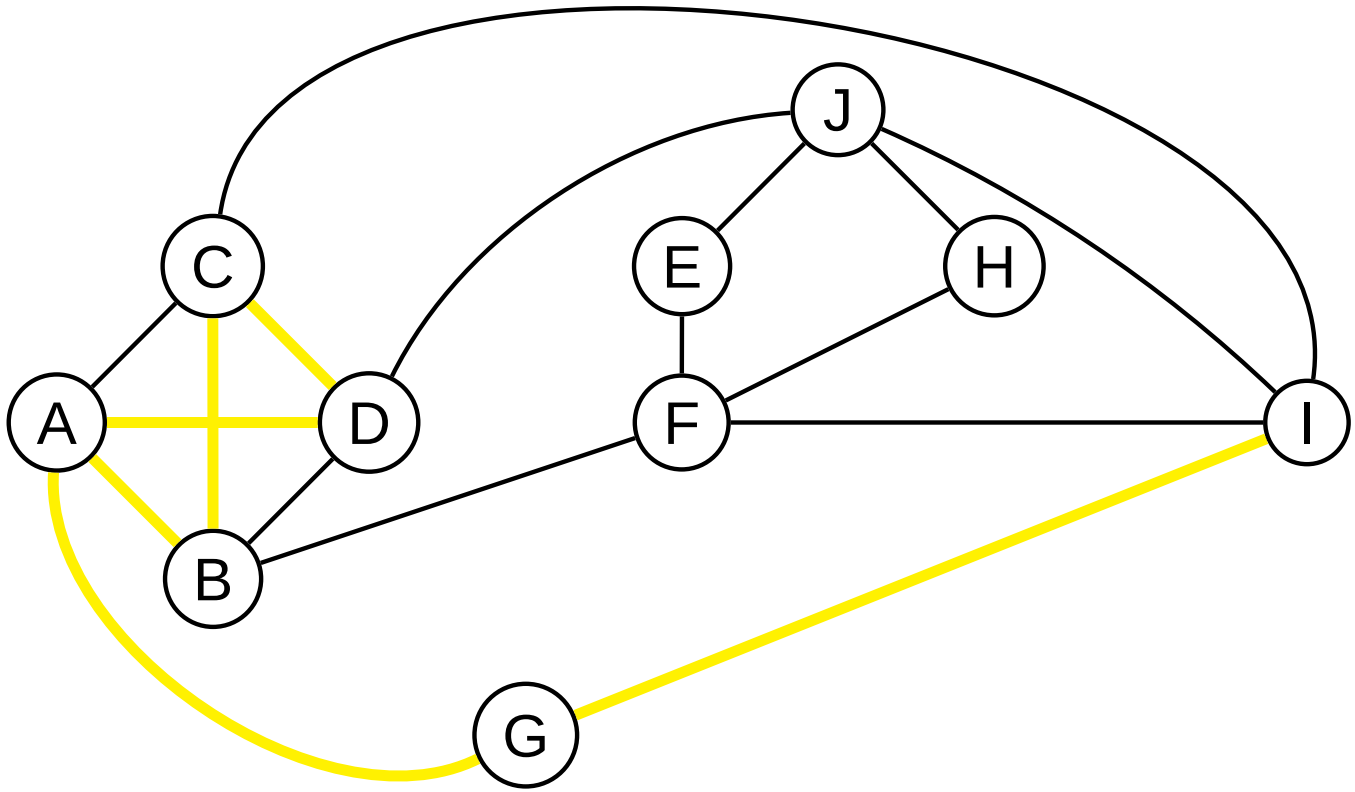
Circuit eulerien



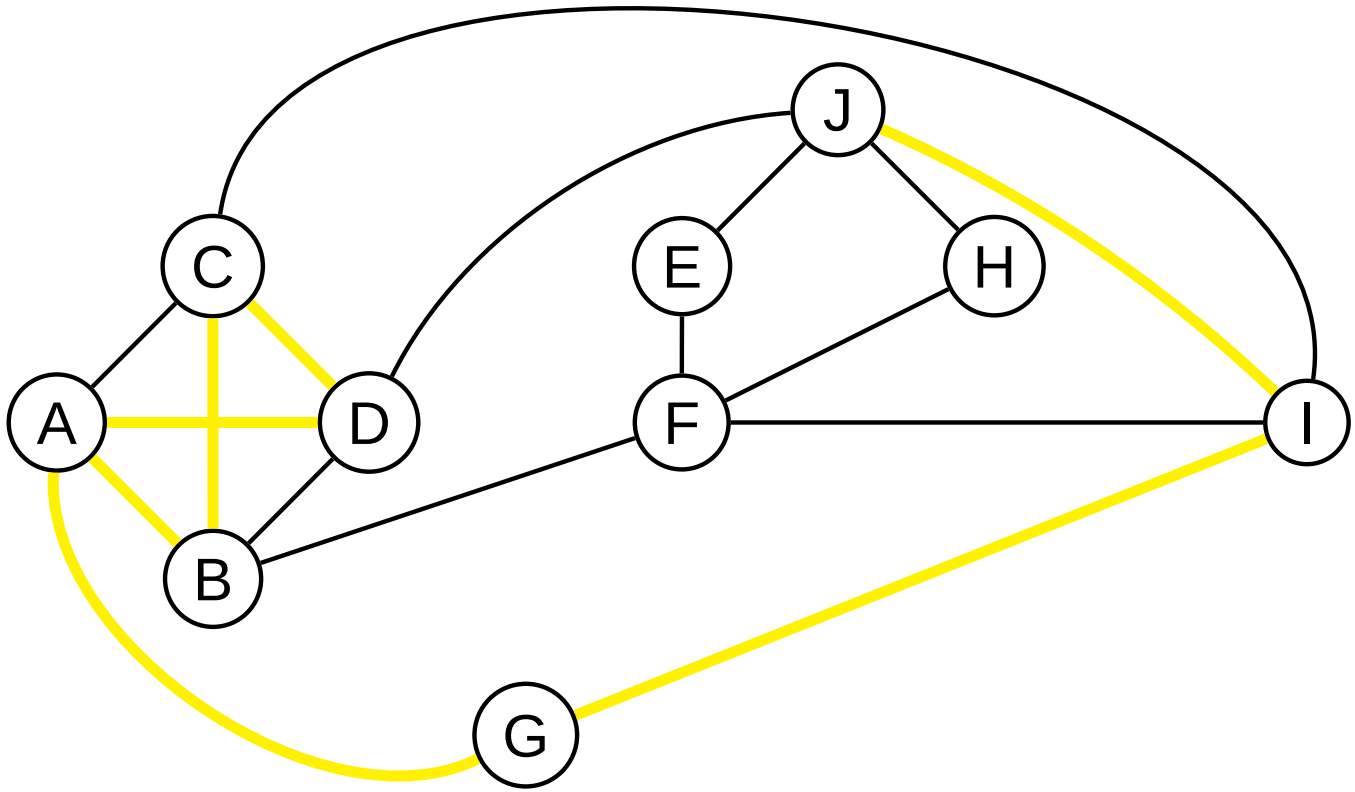
Circuit eulerien



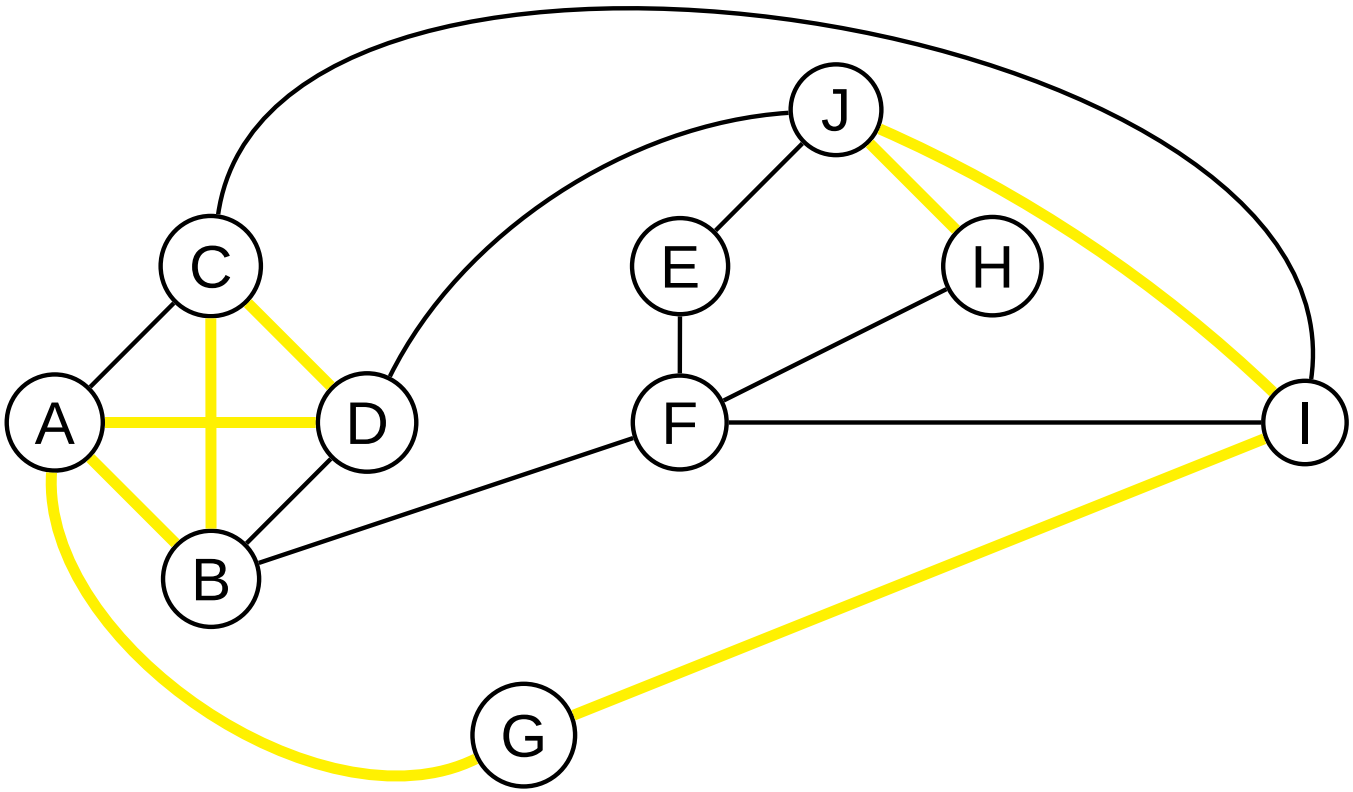
Circuit eulerien



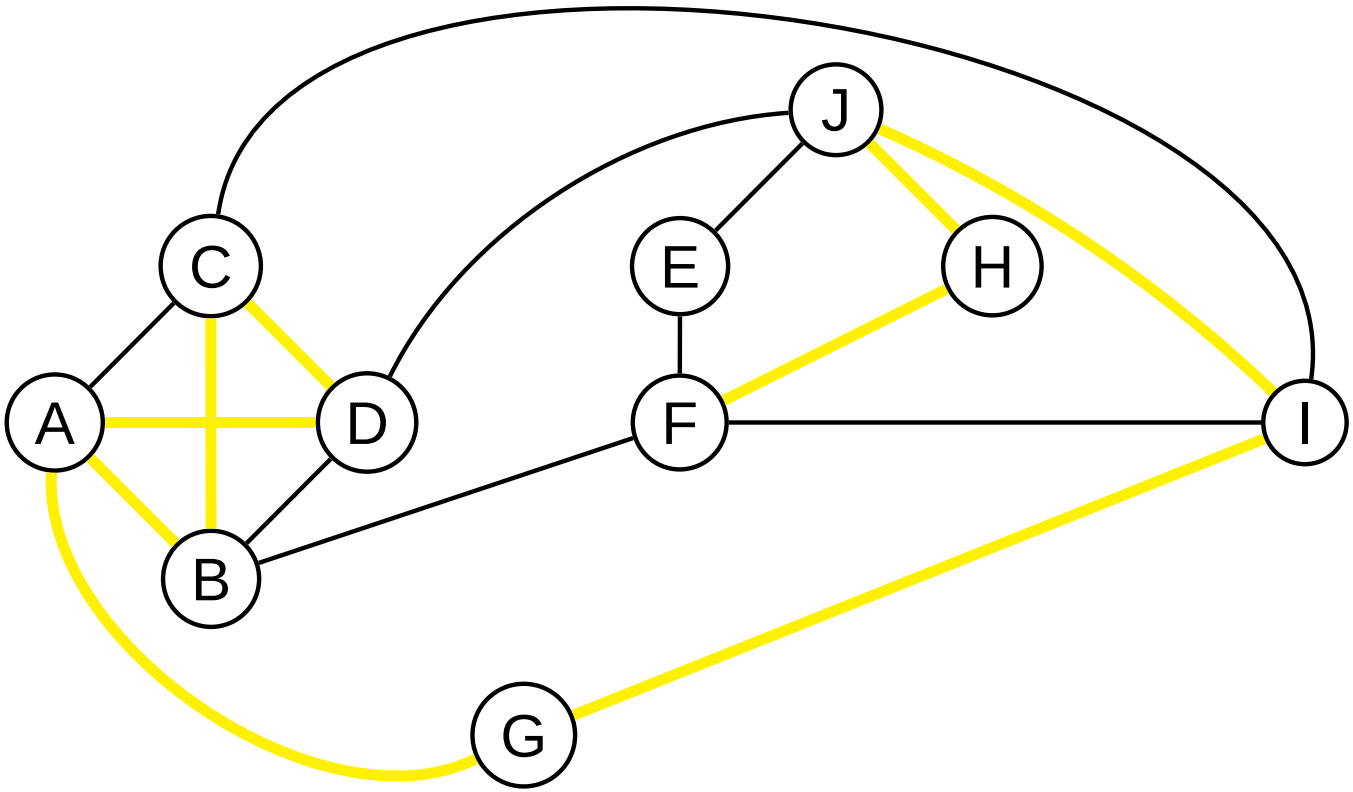
Circuit eulerien



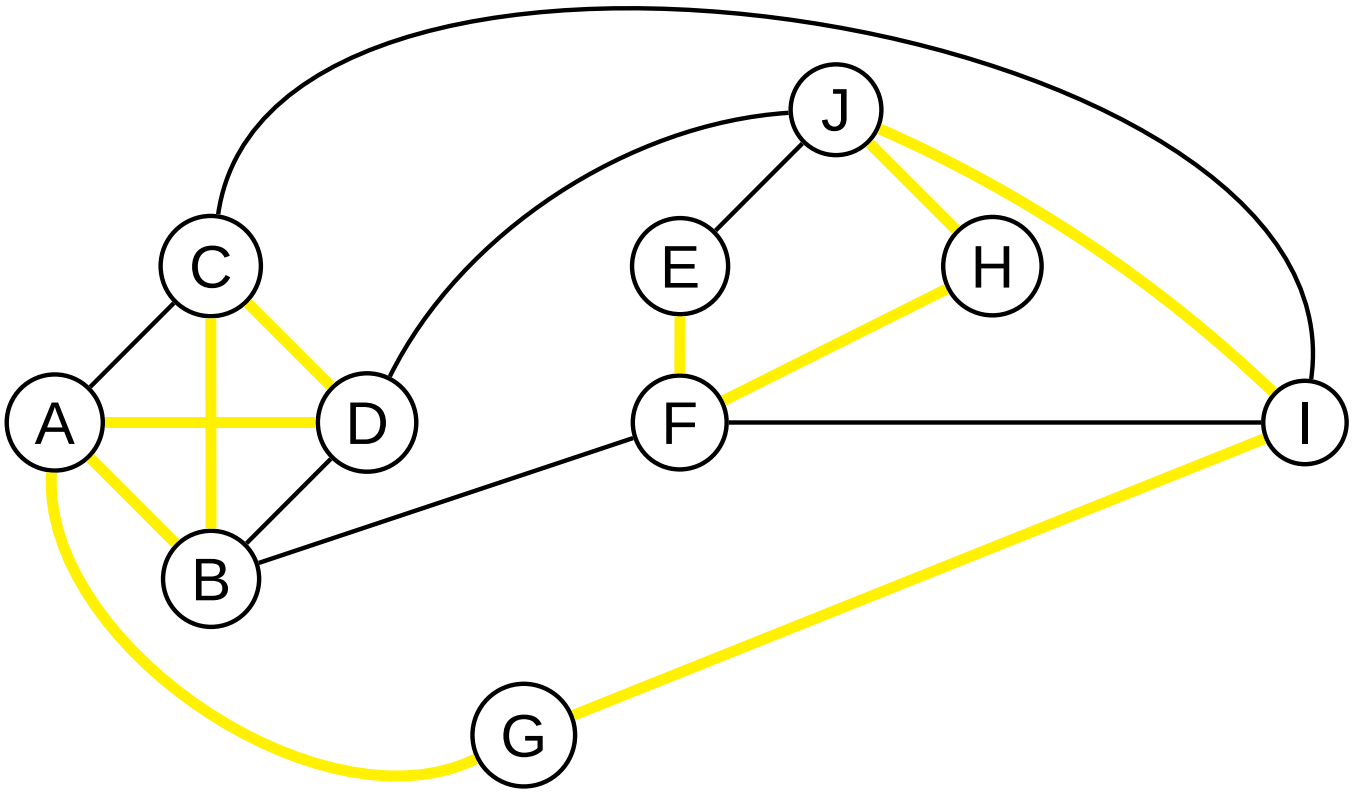
Circuit eulerien



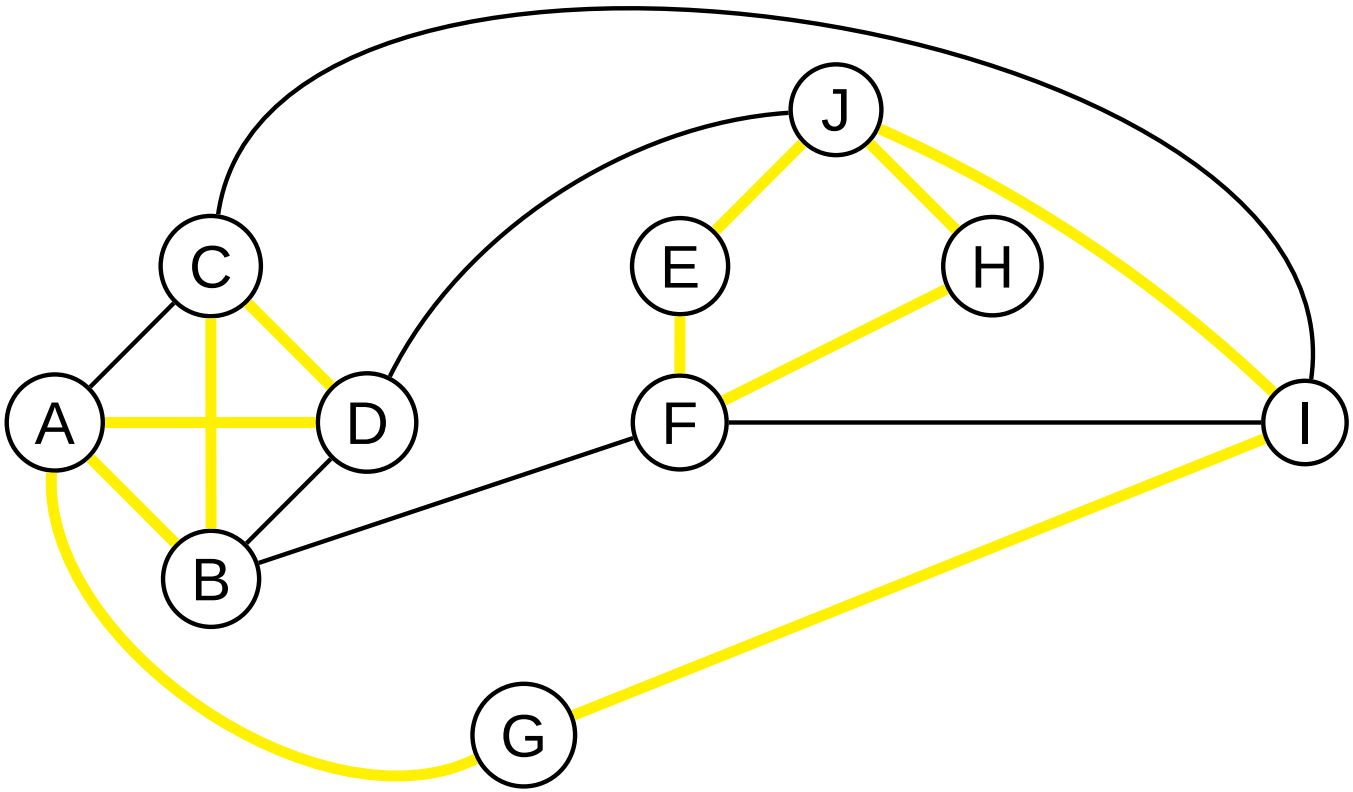
Circuit eulerien



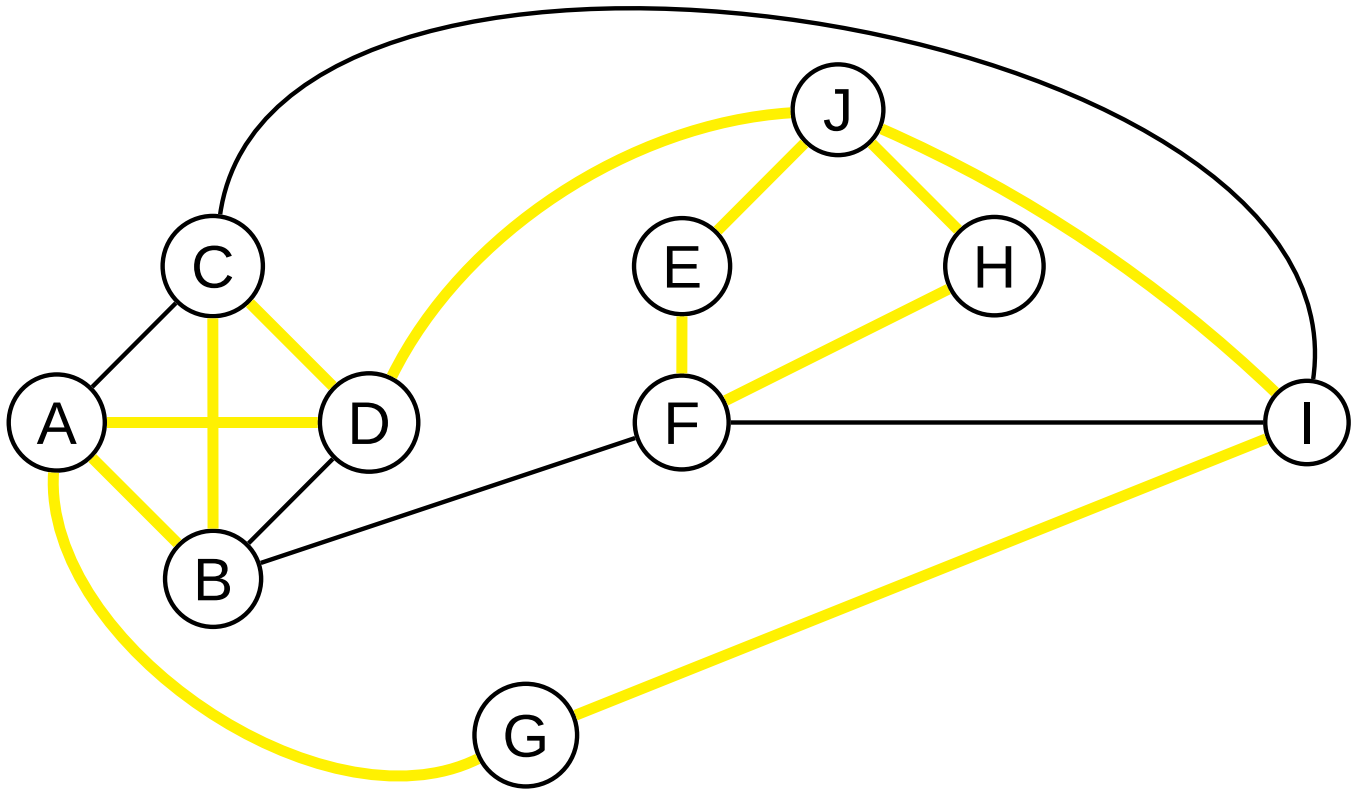
Circuit eulerien



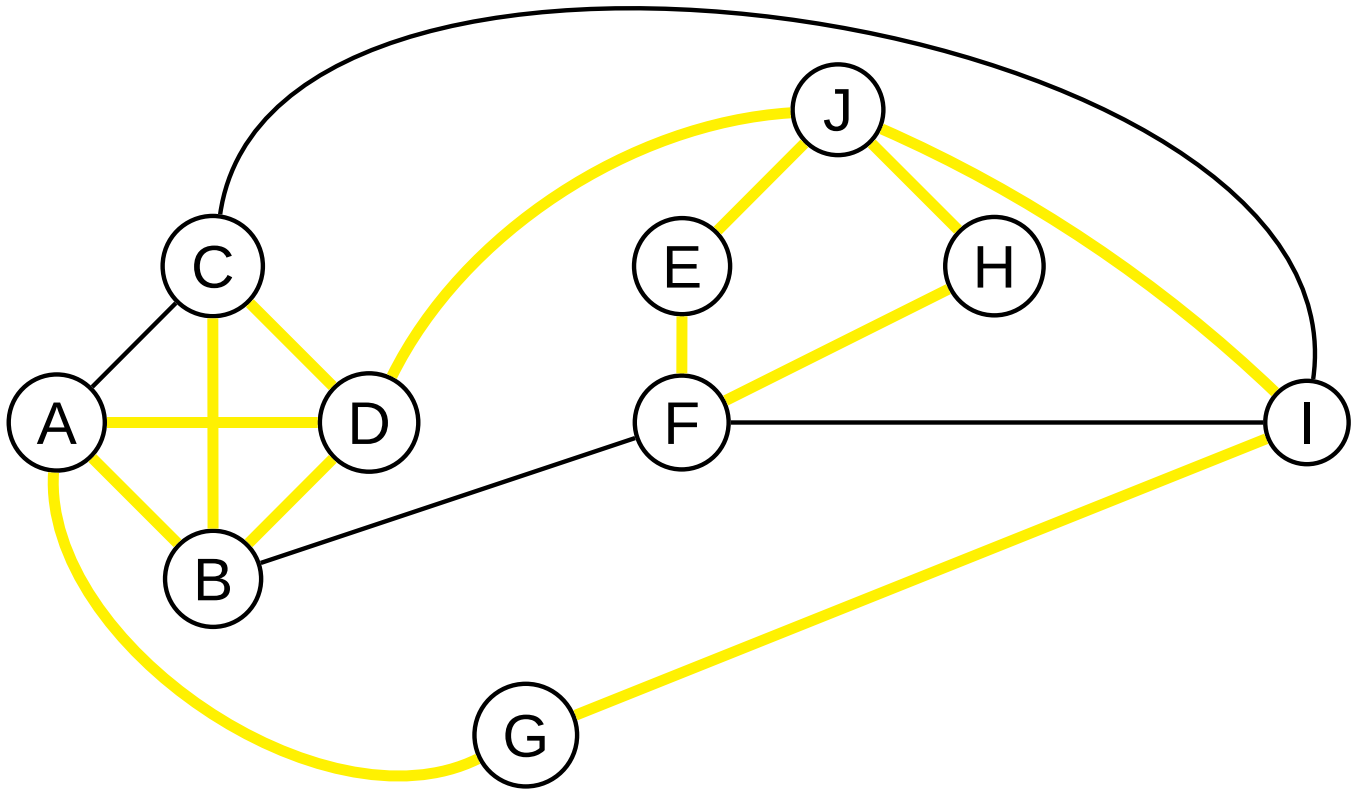
Circuit eulerien



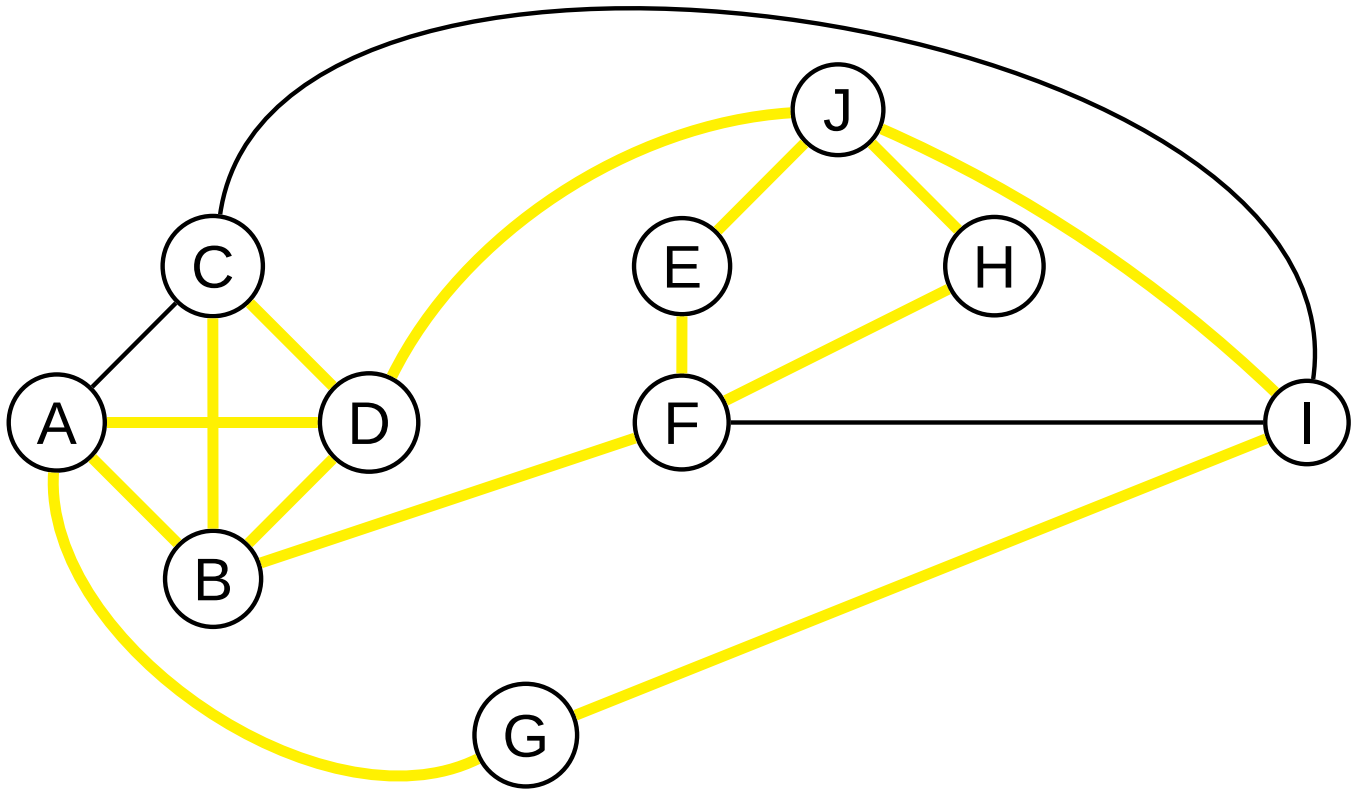
Circuit eulerien



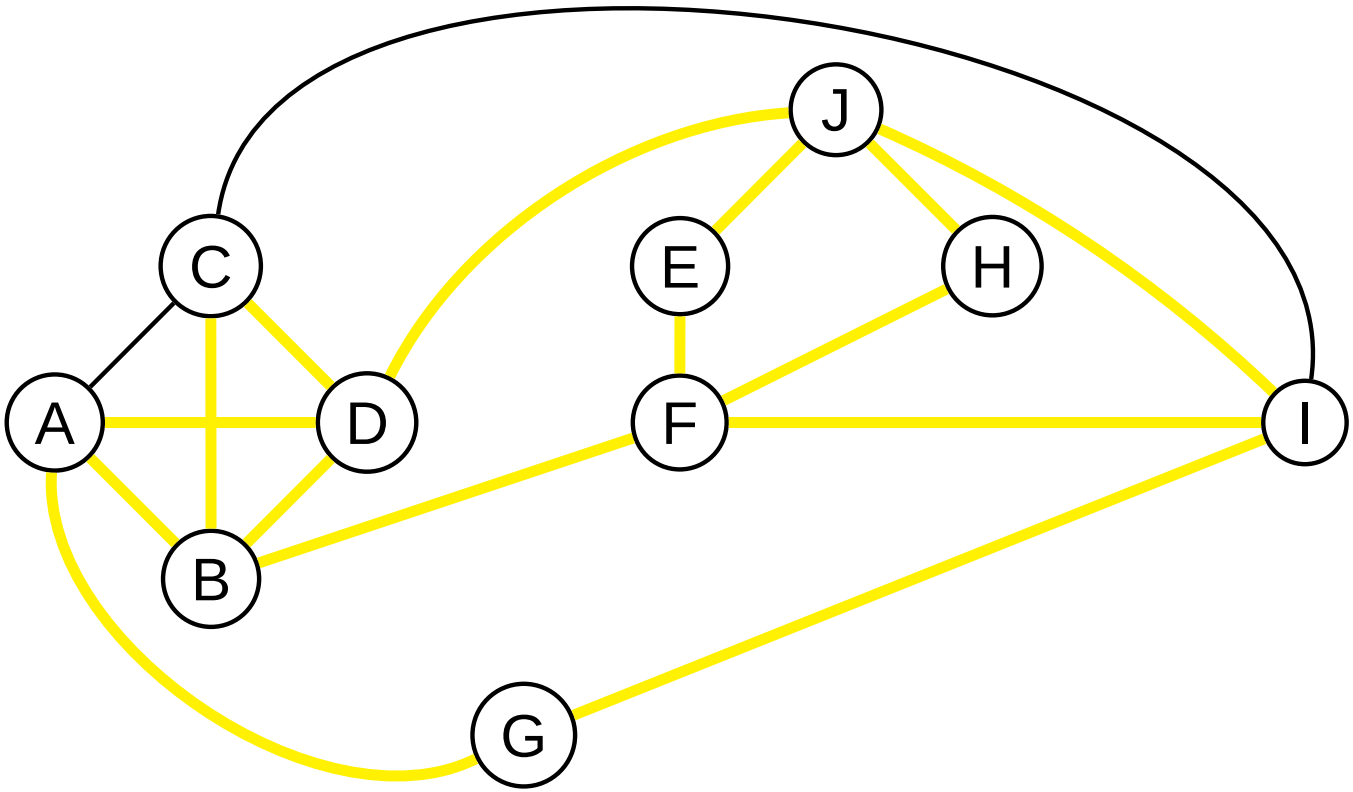
Circuit eulerien



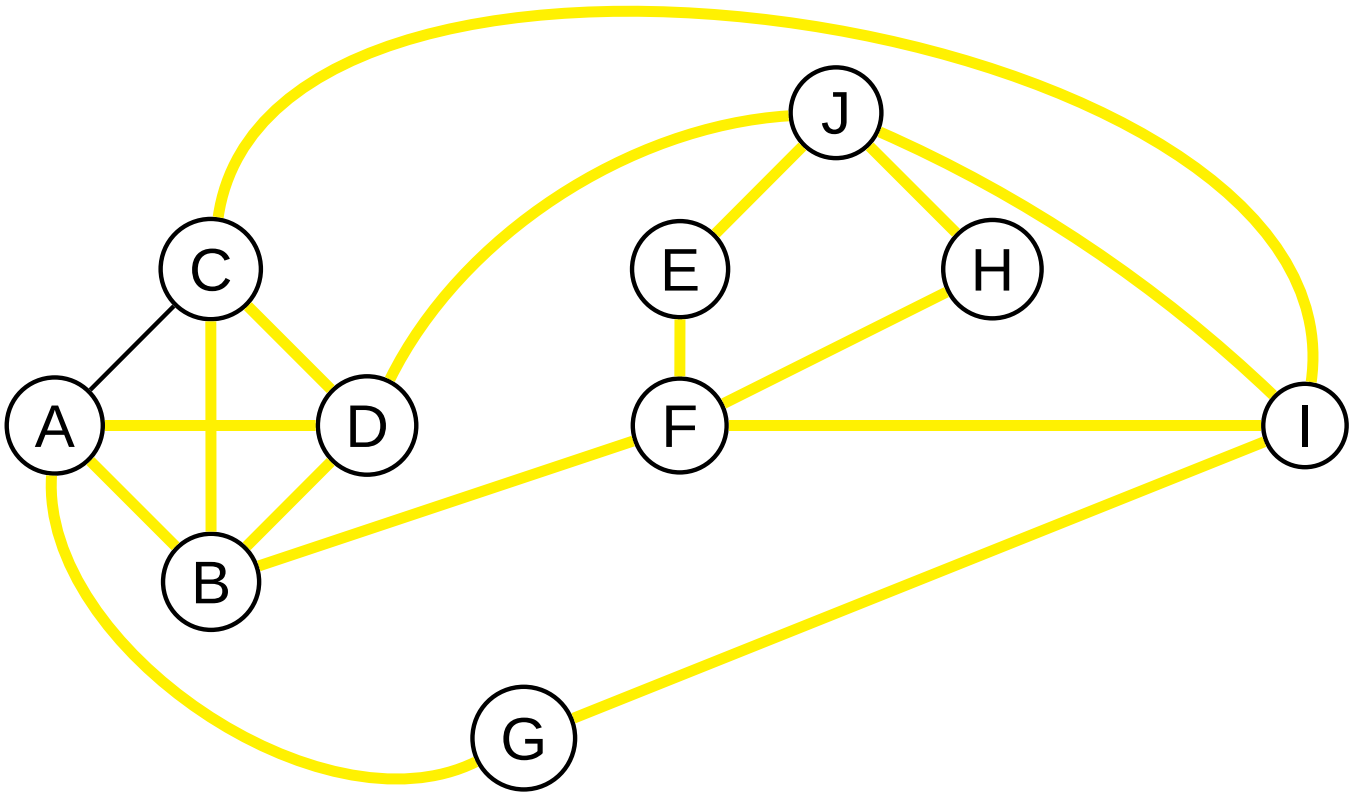
Circuit eulerien



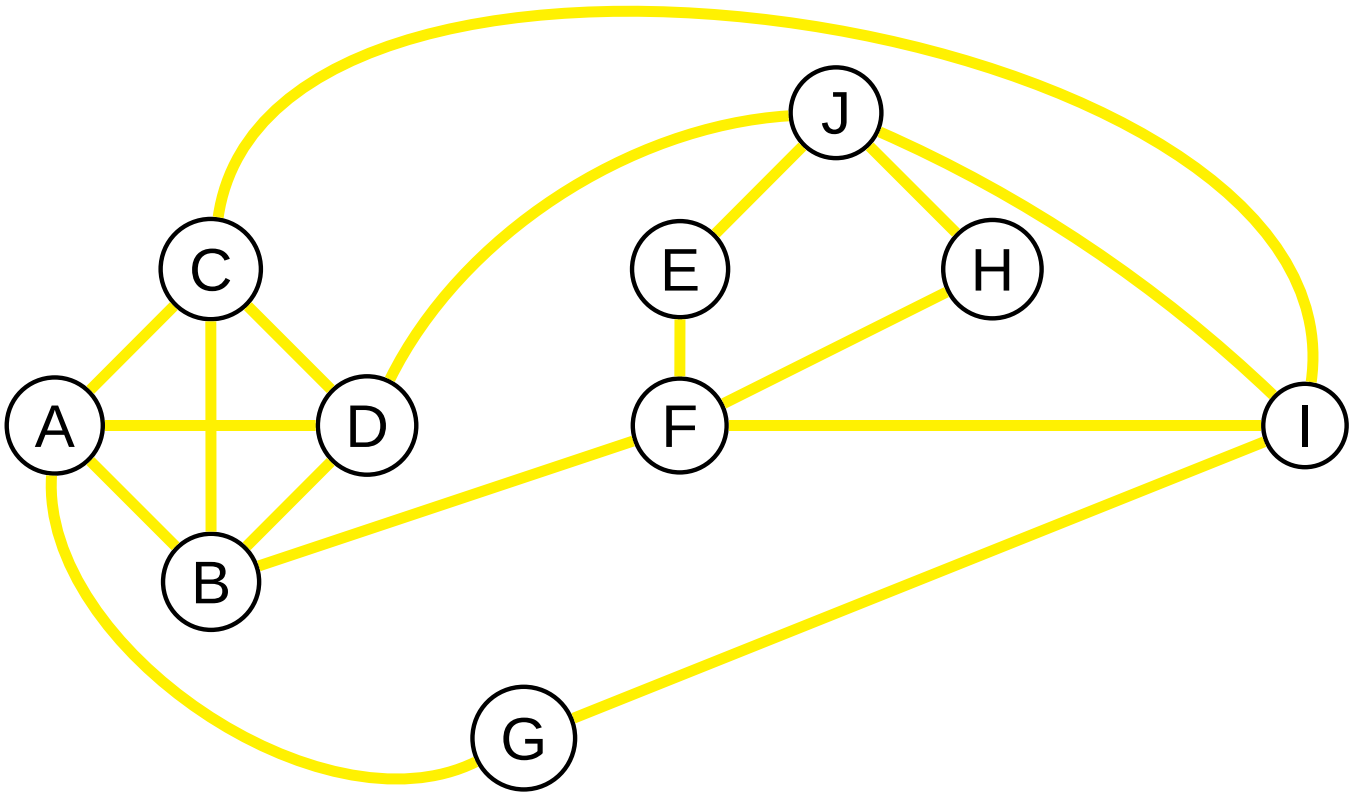
Circuit eulerien



Circuit eulerien



Circuit eulerien



Bilan

Nombre d'étapes pour le circuit Eulerien

besoin de tester le degré de chaque sommet. En tout, $|V|^2$ étapes (au grand maximum... $O(|E|)$ en fait).

Le nombre d'étapes est polynomial en la taille du graphe.

Bilan

Circuit hamiltonien : a priori difficile (exponentiel)
Circuit eulerien : facile (polynomial)

k-colorabilité

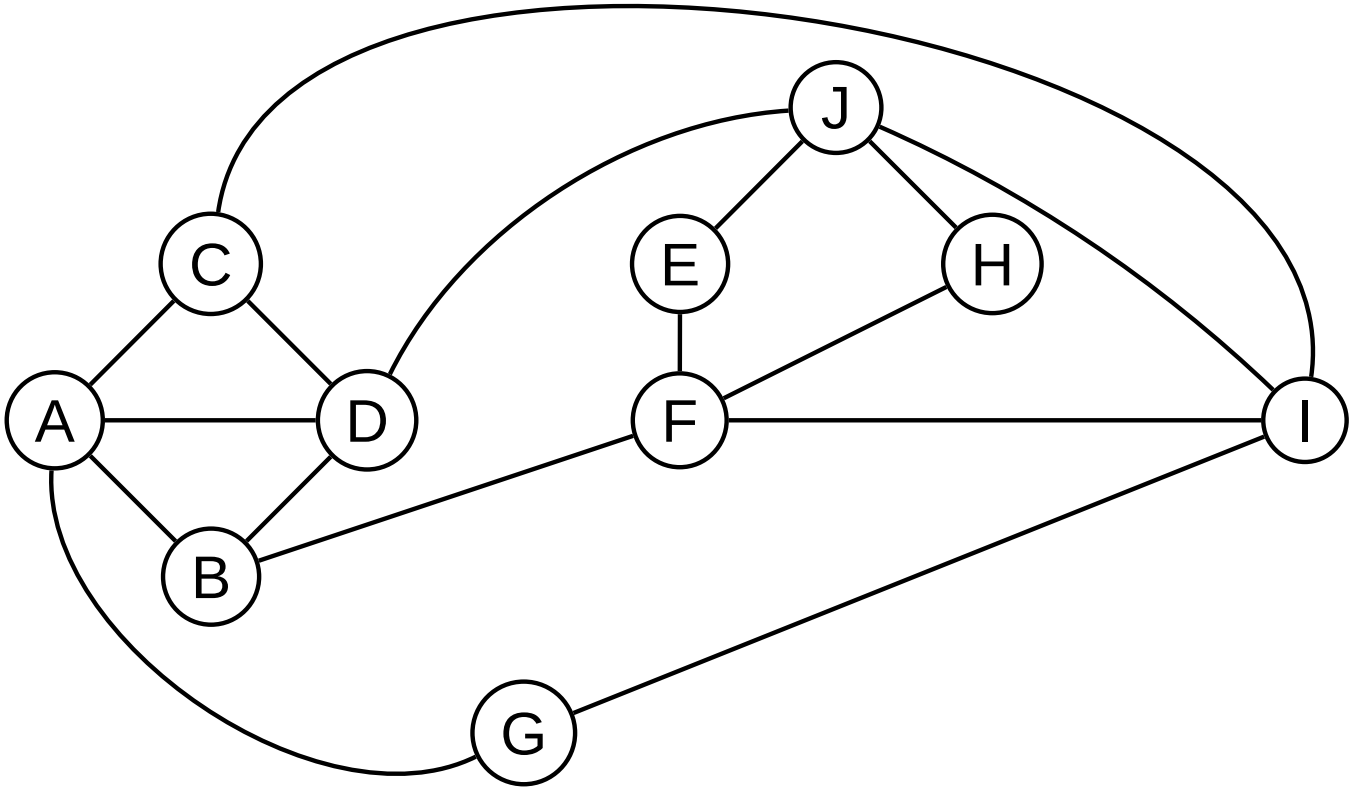
k-colorabilité

Étant donnés un graphe G et un entier k, peut-on colorier les sommets de G avec au plus k couleurs de façon à ce que deux sommets adjacents ne portent jamais la même couleur.

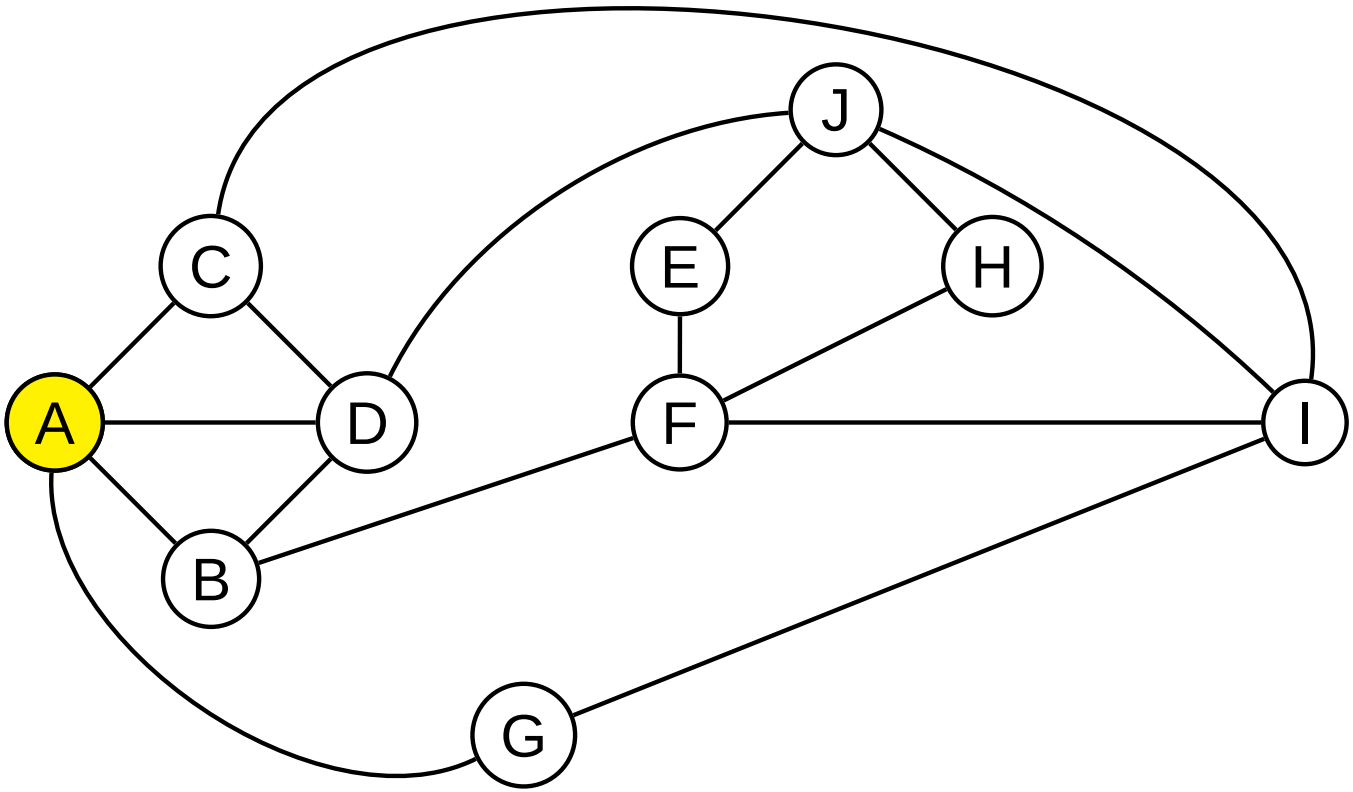
Si le graphe G est une carte et $k \geq 4$, la réponse est toujours oui.

3-colorabilité

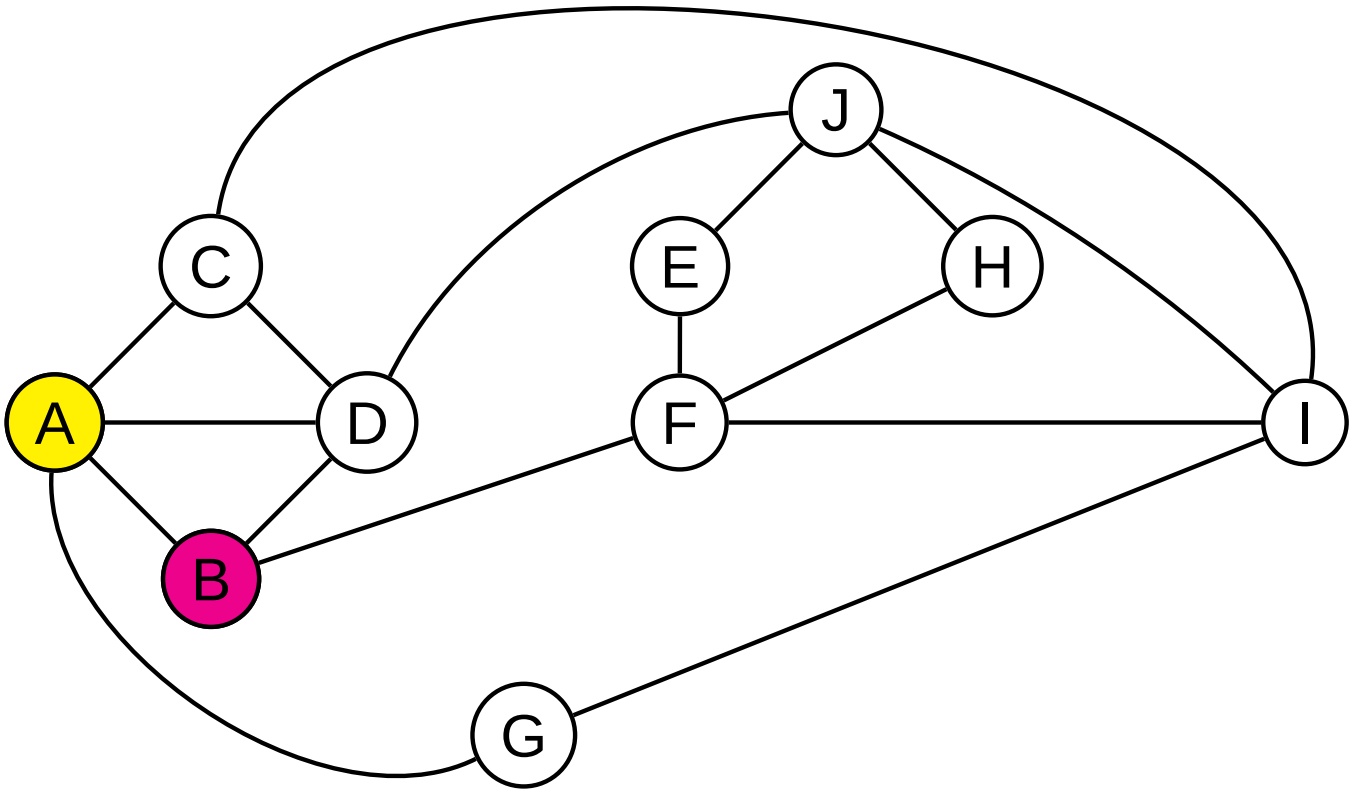
3 couleurs : jaune,bleu,rouge



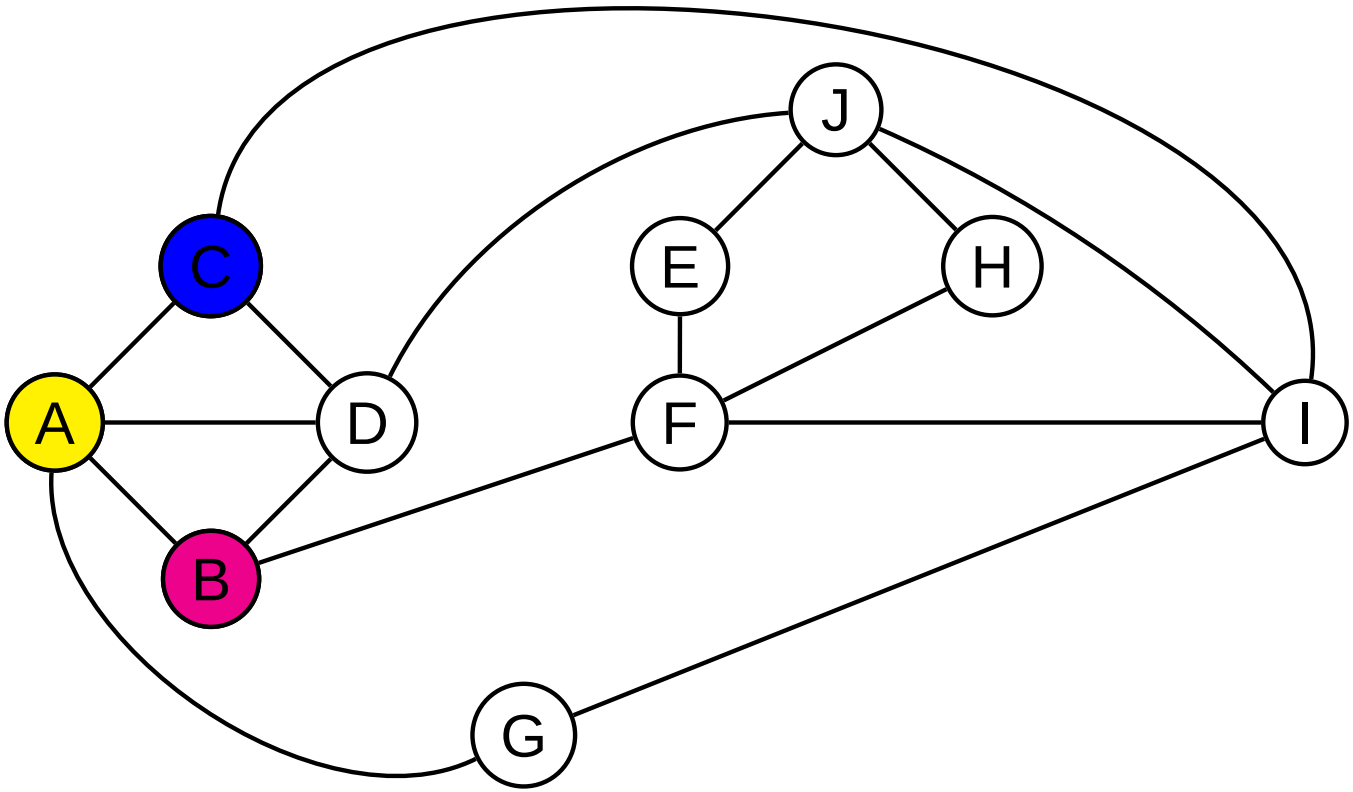
3-colorabilité



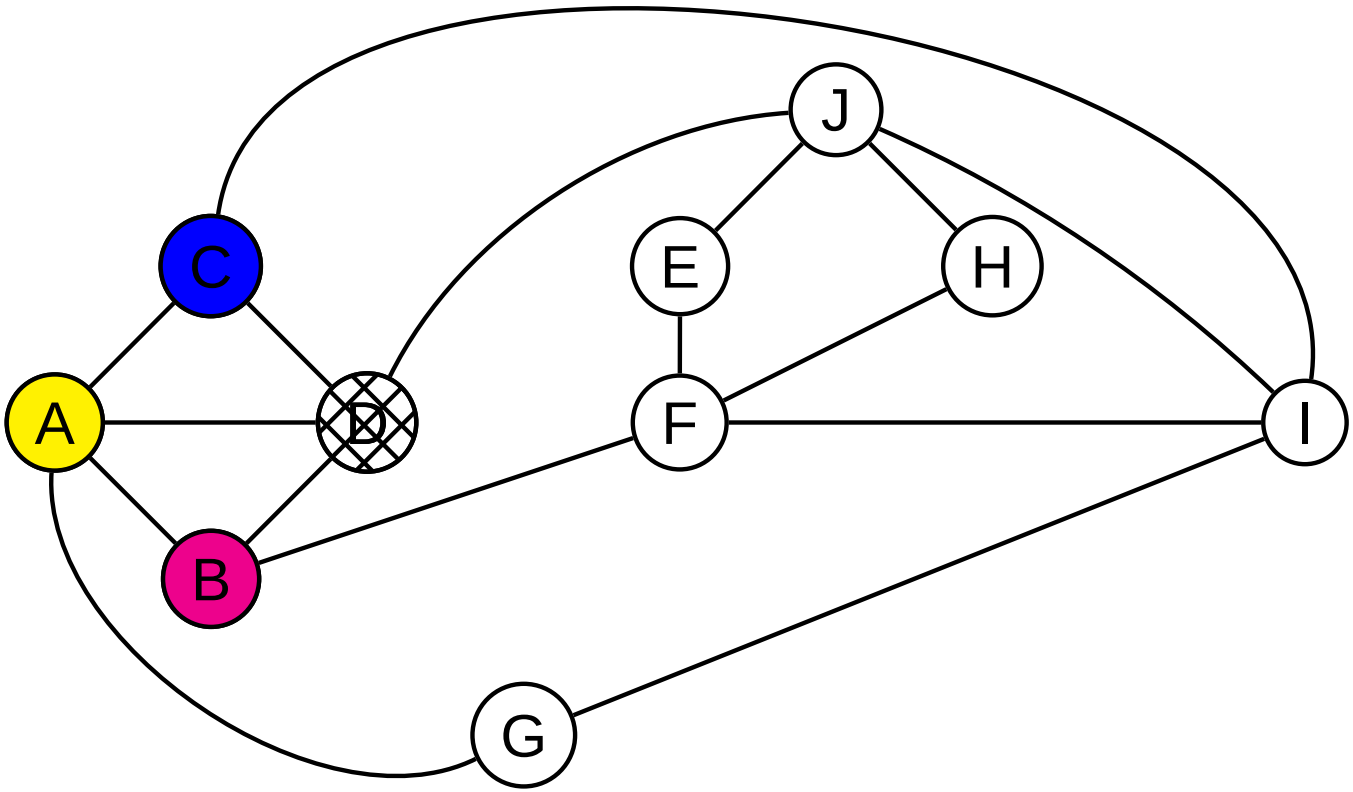
3-colorabilité



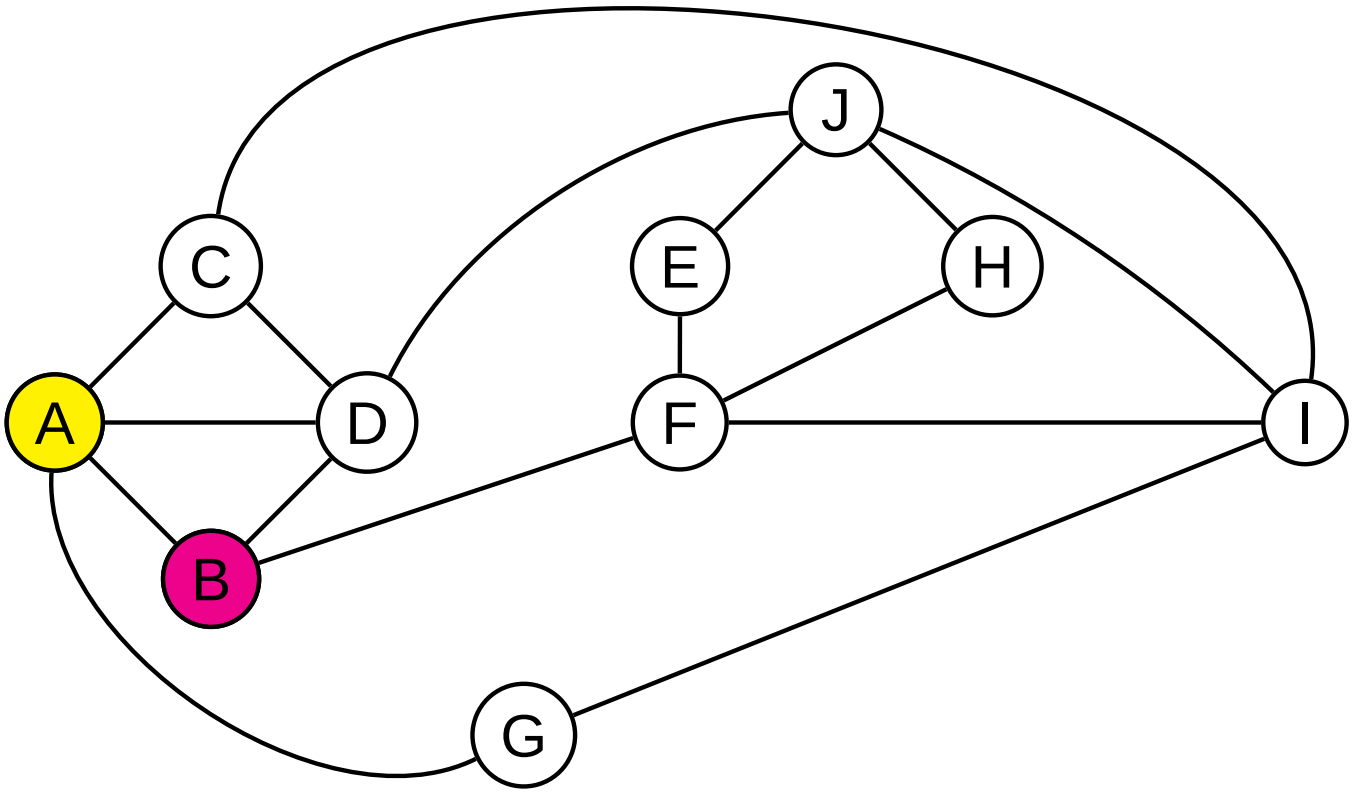
3-colorabilité



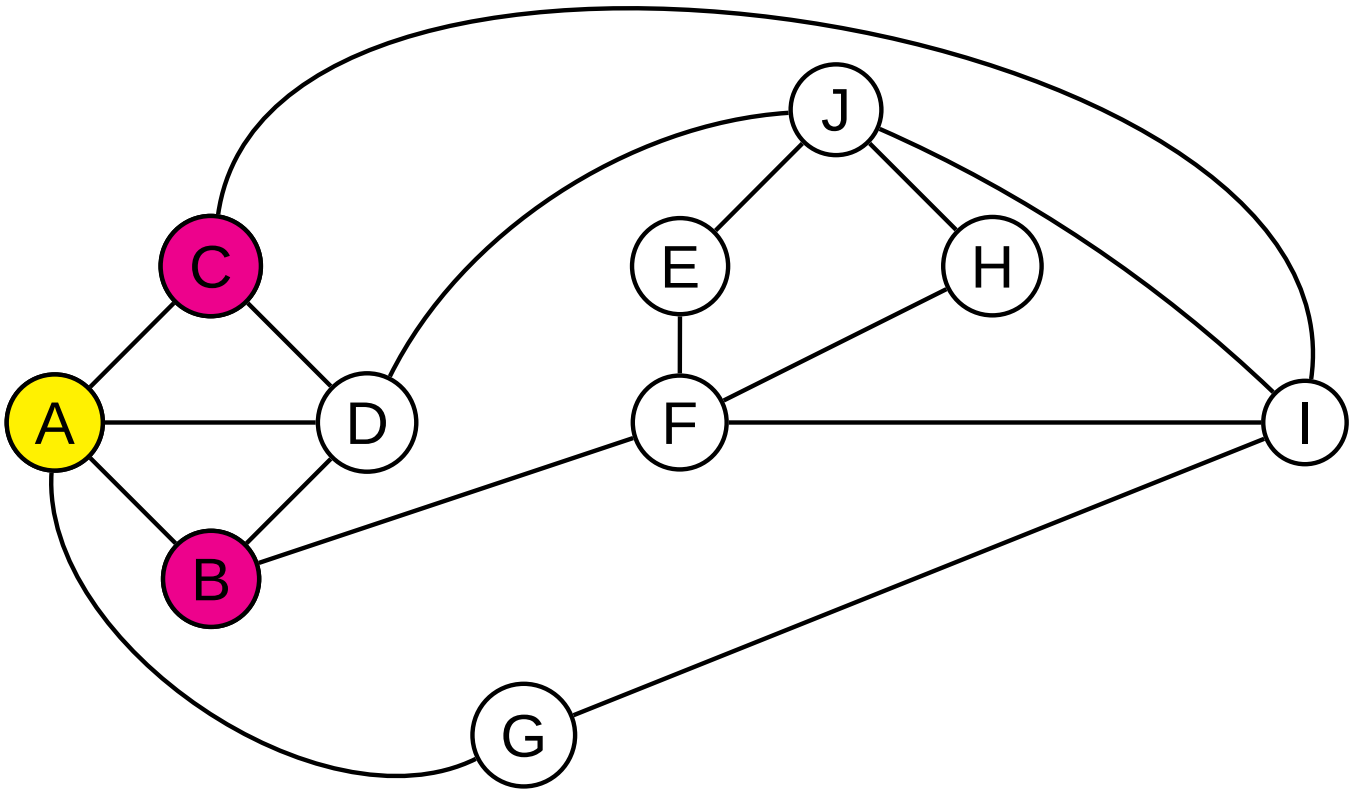
3-colorabilité



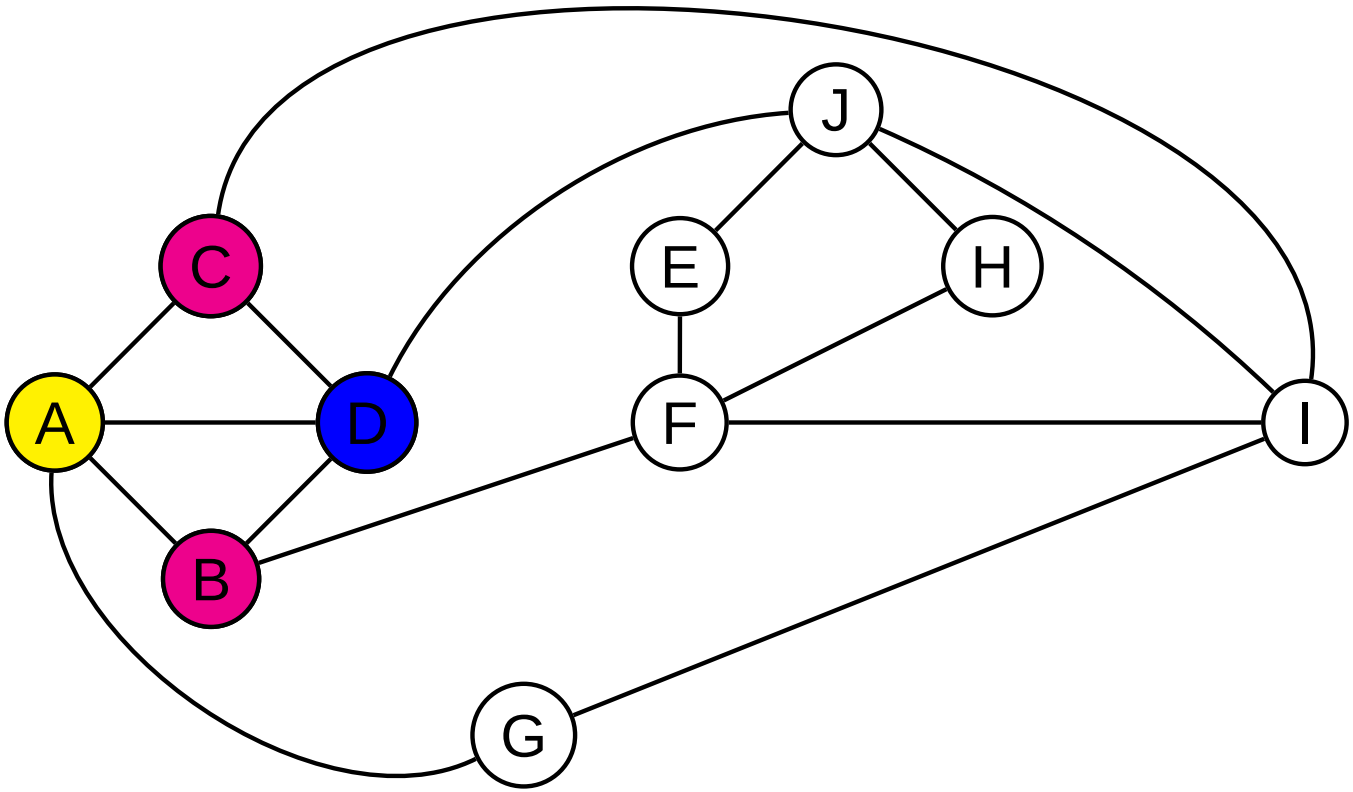
3-colorabilité



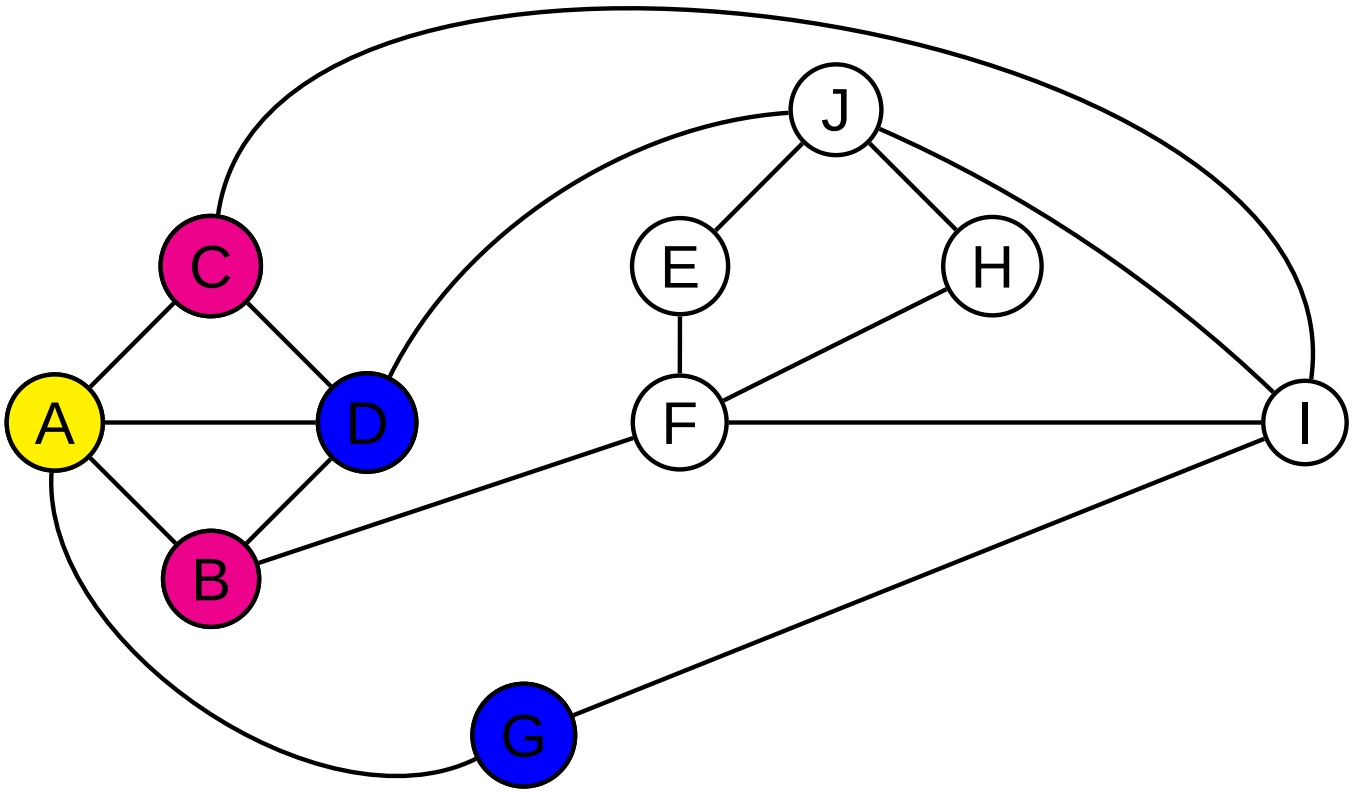
3-colorabilité



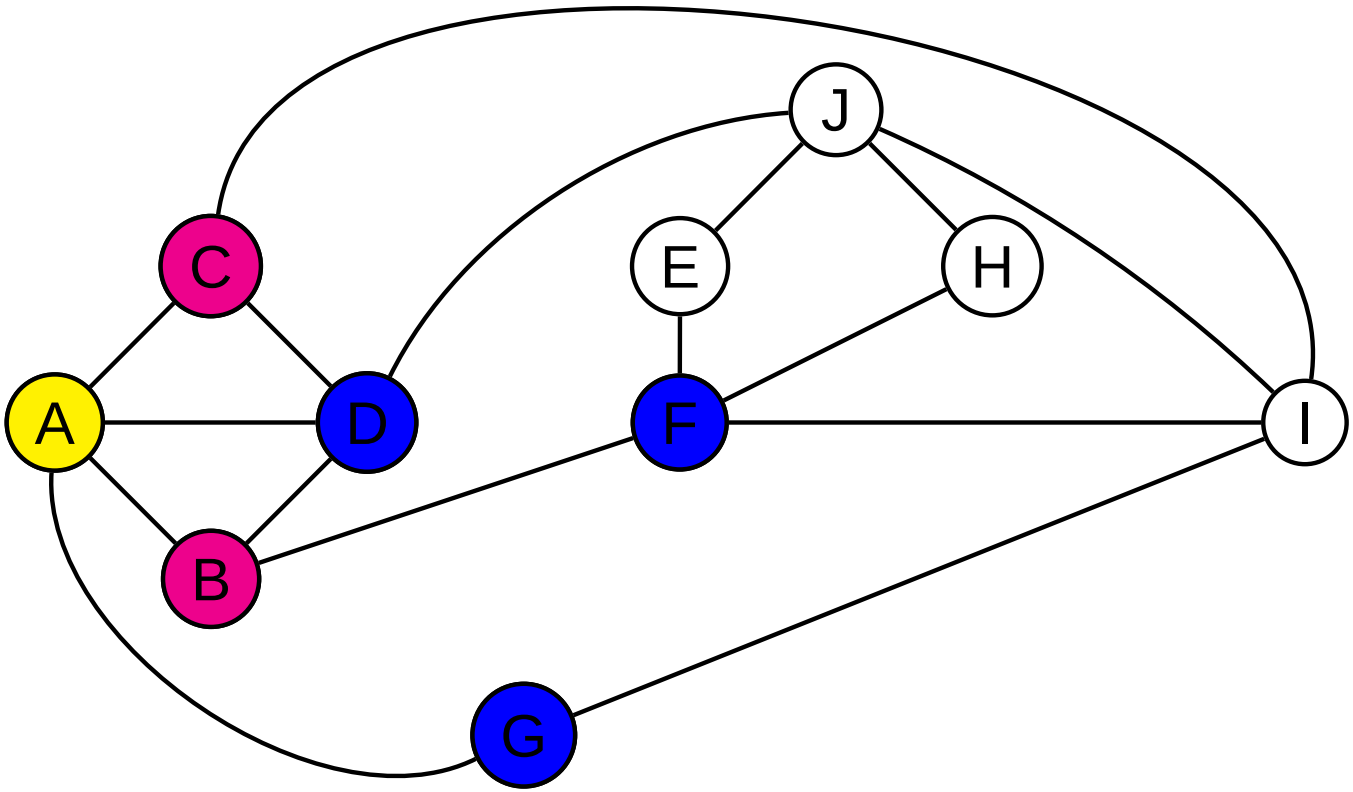
3-colorabilité



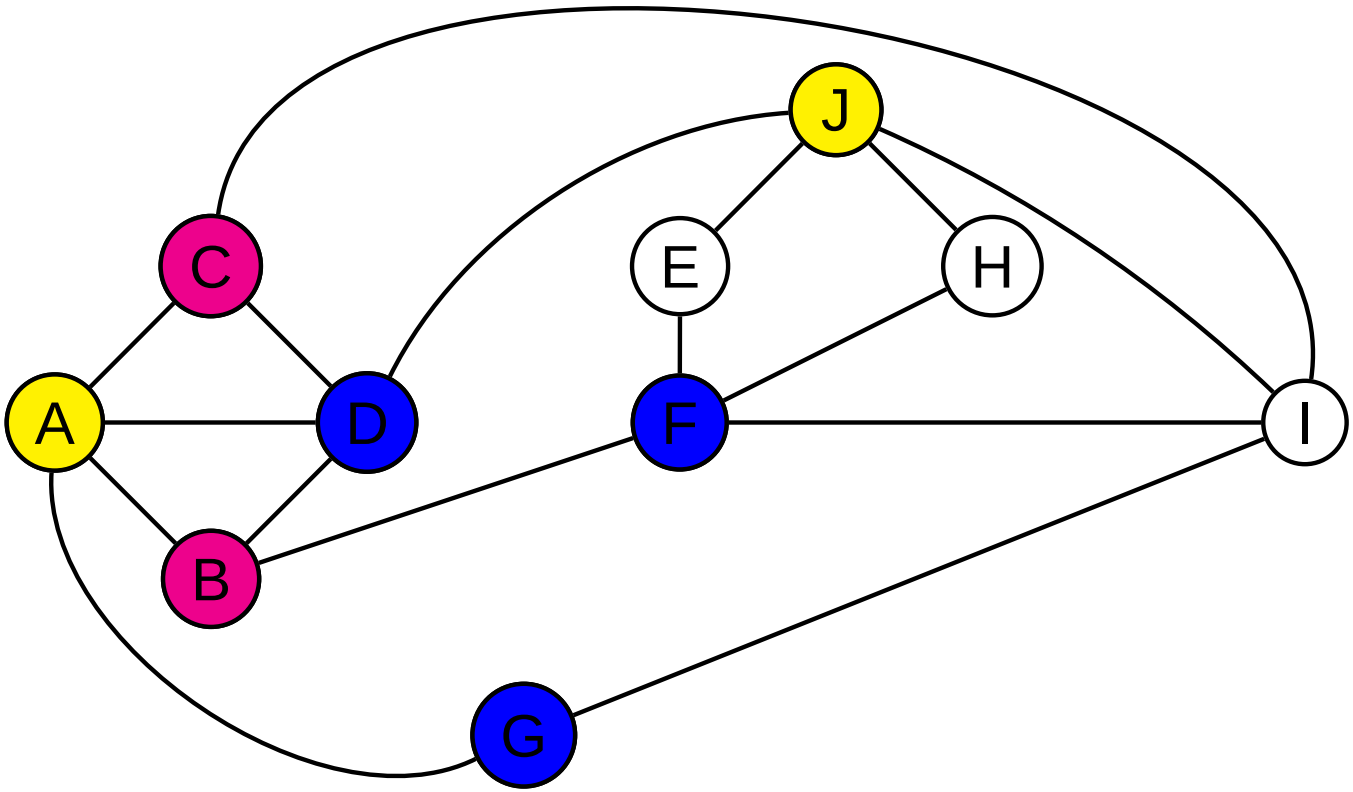
3-colorabilité



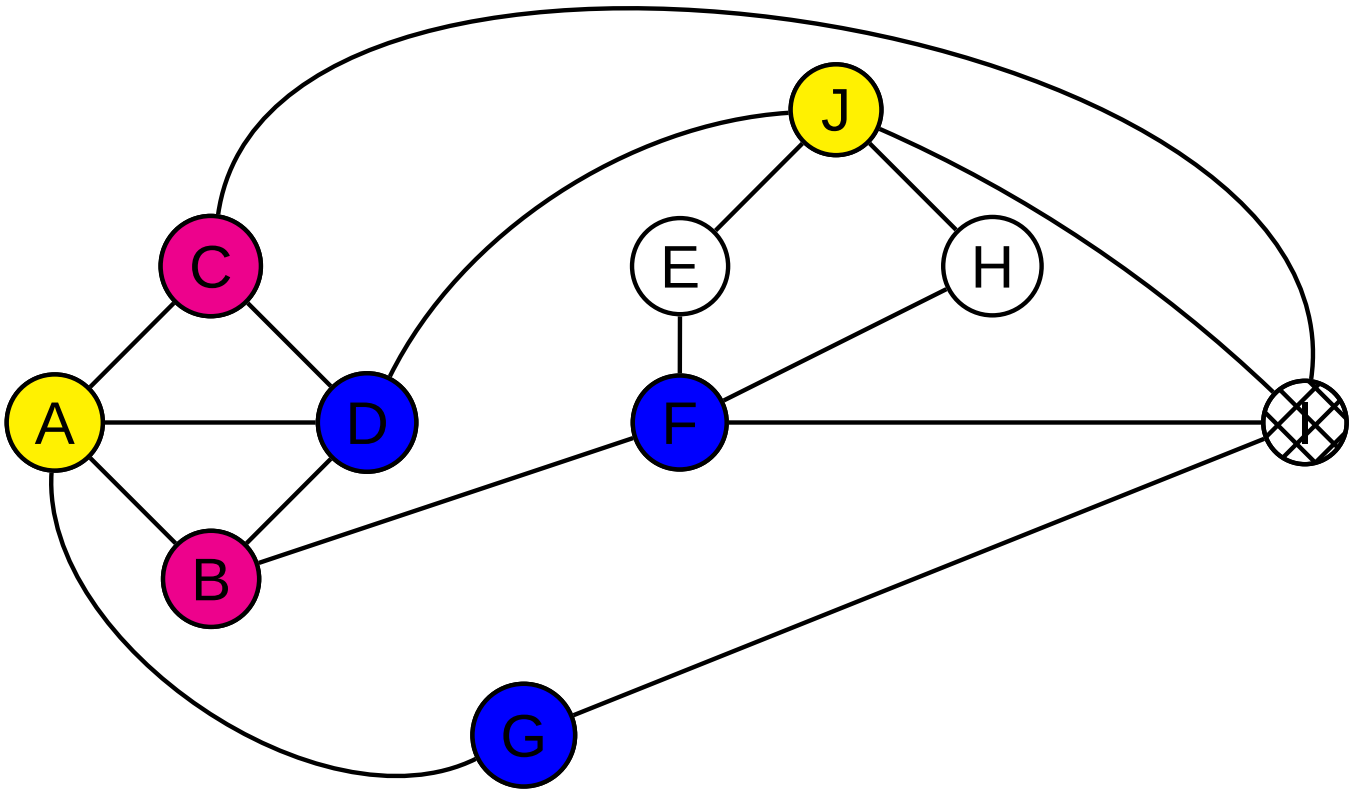
3-colorabilité



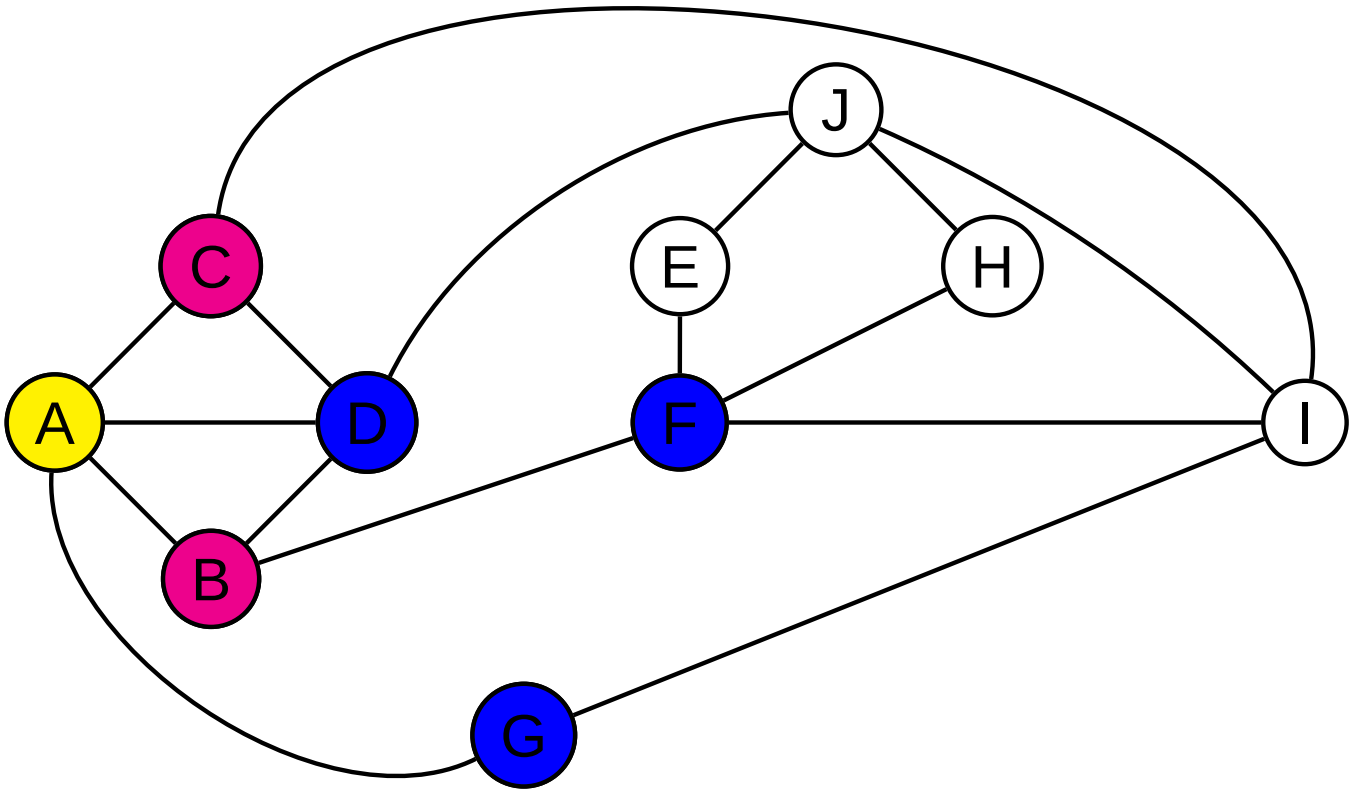
3-colorabilité



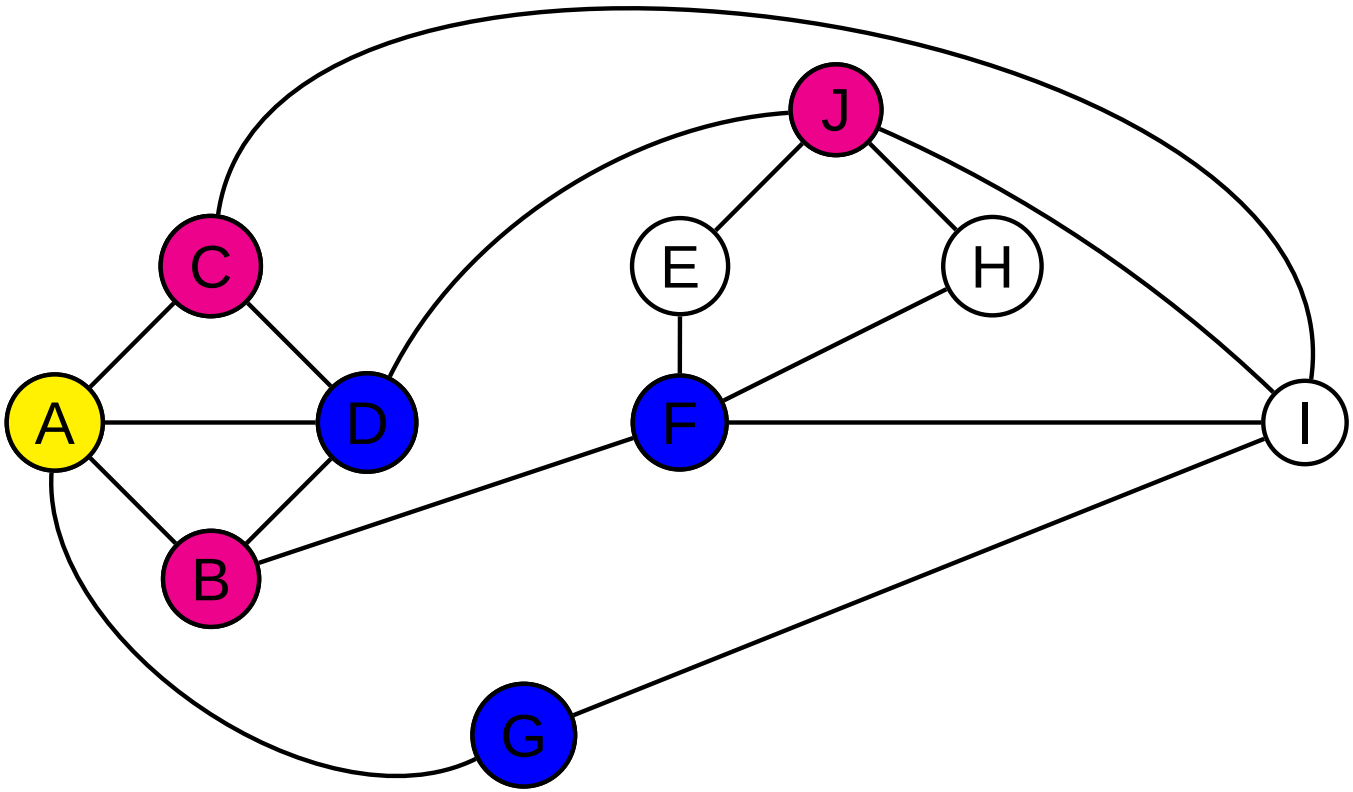
3-colorabilité



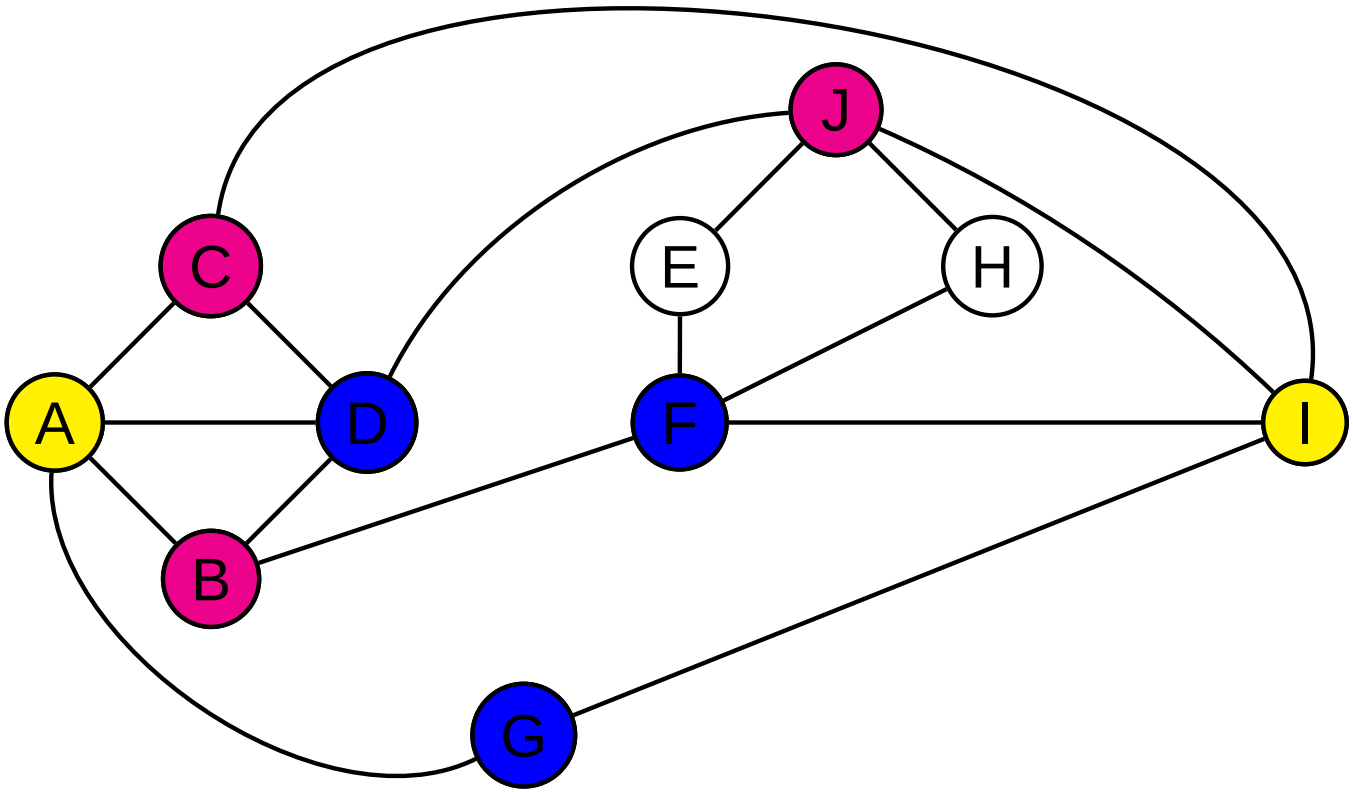
3-colorabilité



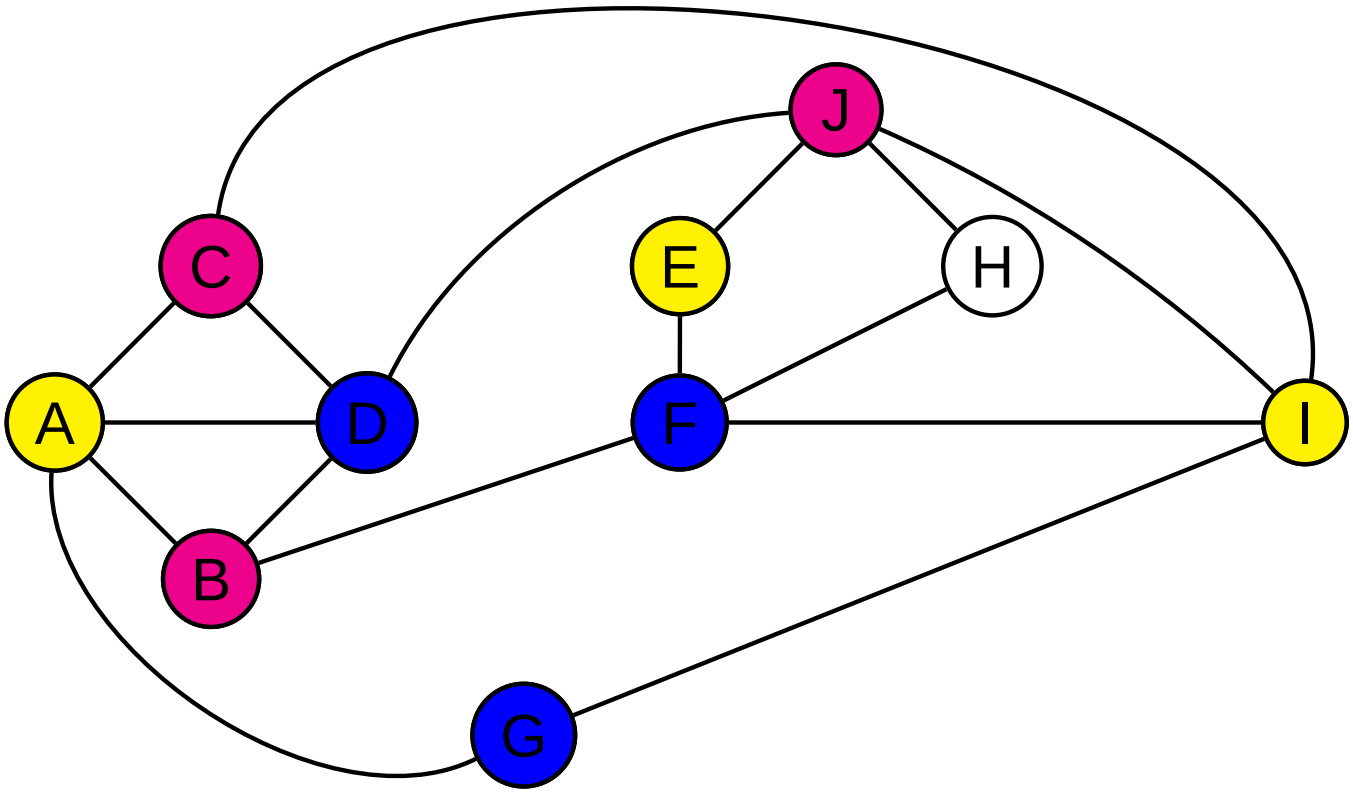
3-colorabilité



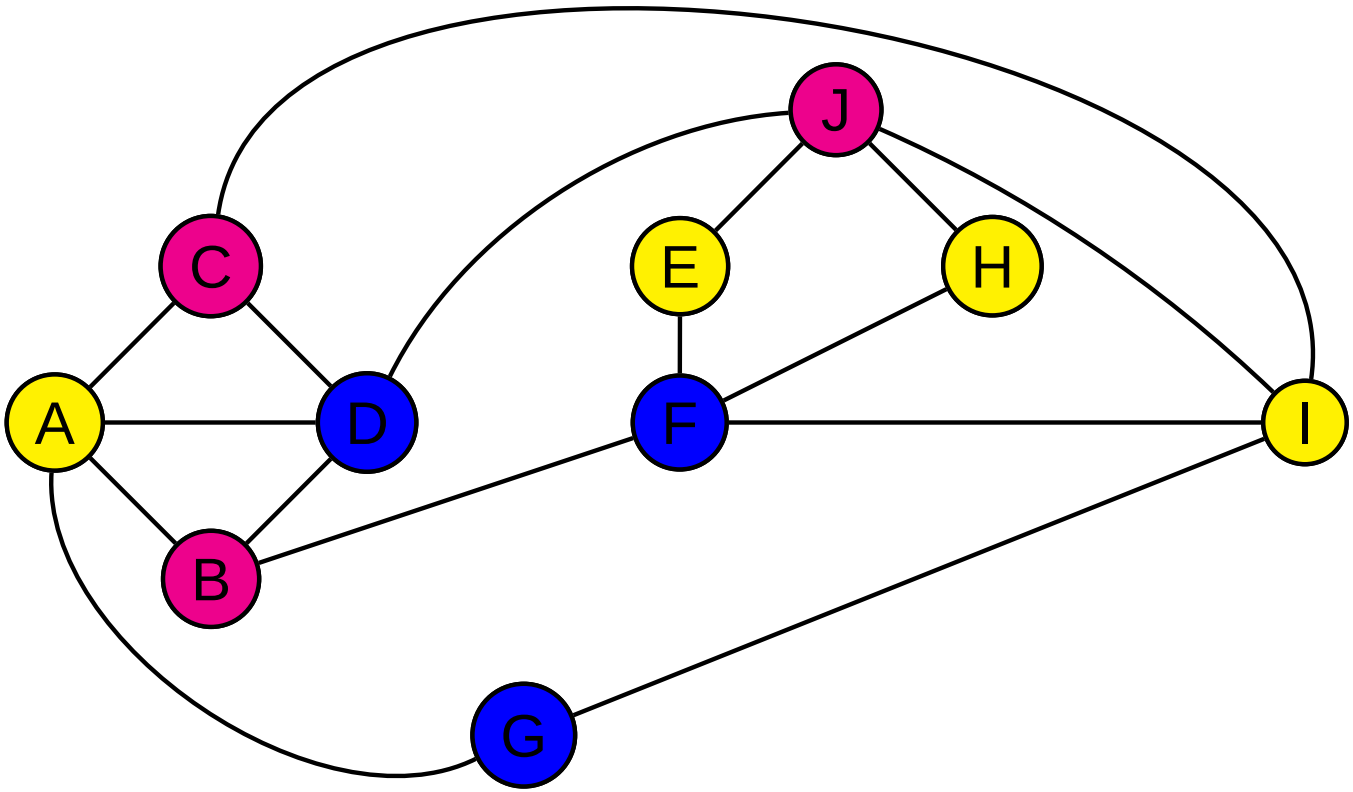
3-colorabilité



3-colorabilité

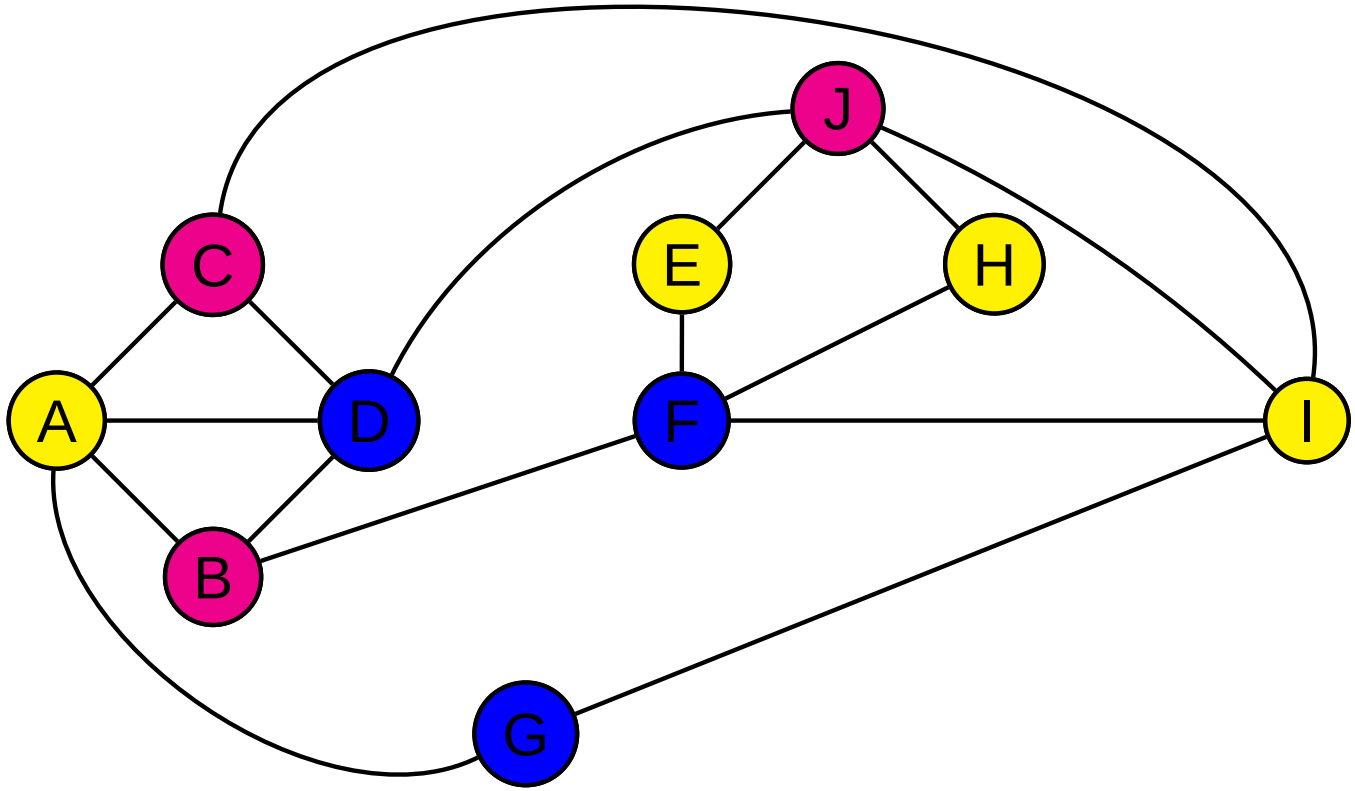


3-colorabilité



3-colorabilité : bilan

A nouveau quelques retours en arrière. Tester toutes les possibilités : 3^n où n est le nombre de sommets. Pas d'algorithme connu avec un nombre polynomial d'étapes.



Colorabilité des arêtes

Indice Chromatique

Étant donnés un graphe G et un entier k, peut-on colorier les arêtes de G avec au plus k couleurs de façon à ce que deux arêtes incidentes ne portent jamais la même couleur.
Nombre minimal de couleurs : indice chromatique

$\Delta(G)$: degré du graphe (nombre maximal de voisins admis par un sommet)

Vizing (1964)

L'indice chromatique d'un graphe G est soit $\Delta(G)$ soit $\Delta(G) + 1$.

Colorabilité des arêtes

Problème facile si $k < \Delta(G)$ ou $k > \Delta(G) + 1$
En route vers un algorithme efficace ? d'autant plus que :

Erdös, Wilson (1977)

Lorsque n est grand, presque tous les graphes sont d'indice $\Delta(G)$

Colorabilité des arêtes

Problème facile si $k < \Delta(G)$ ou $k > \Delta(G) + 1$
En route vers un algorithme efficace ? d'autant plus que :

Erdös, Wilson (1977)

Lorsque n est grand, presque tous les graphes sont d'indice $\Delta(G)$

Mais :

Holyer (1981)

Sauf surprise, il n'existe pas d'algorithme polynomial pour décider si un graphe est d'indice $\Delta(G)$ ou $\Delta(G) + 1$.

Polynomial vs exponentiel

Quelle différence ?

n	10	100	1000
n^2	100	10^4	10^6
n^5	10^5	10^{10}	10^{15}
2^n	2^{10}	$> 10^{25}$	$> 10^{250}$

PC actuels : entre 1 et 4 milliards d'opérations par seconde

Machines les plus rapides (-> Petaflops) : 10^{15} (1 million de milliards d'opérations par seconde).

Nombre d'atomes dans l'univers observable : 10^{80}

Problèmes artificiels ou pratiques ?

Problématique naturelle en mathématiques, mais aussi..

- Bases de données
- Sécurité (cryptographie...)
- Algorithmique des réseaux
- Intelligence artificielle (raisonnement, apprentissage)
- Bio-informatique
- Économie (systèmes électoraux, équilibres)
- Jeux
- Ordonnancement (planification de tâches)
- Optimisation

Le temps polynomial

Le temps polynomial **P**

Un problème de décision A est dans **P** s'il est résoluble (i.e. décidable) par un algorithme polynomial.

Proposition comme classe de problèmes faciles ou raisonnables : Cobham (1964), Edmonds (1965), Rabin (1966)

Le temps polynomial

Le temps polynomial **P**

Un problème de décision A est dans **P** s'il est résoluble (i.e. décidable) par un algorithme polynomial.

Proposition comme classe de problèmes faciles ou raisonnables : Cobham (1964), Edmonds (1965), Rabin (1966)

Attention, **P** $\neq$ "facile" de plusieurs points de vue

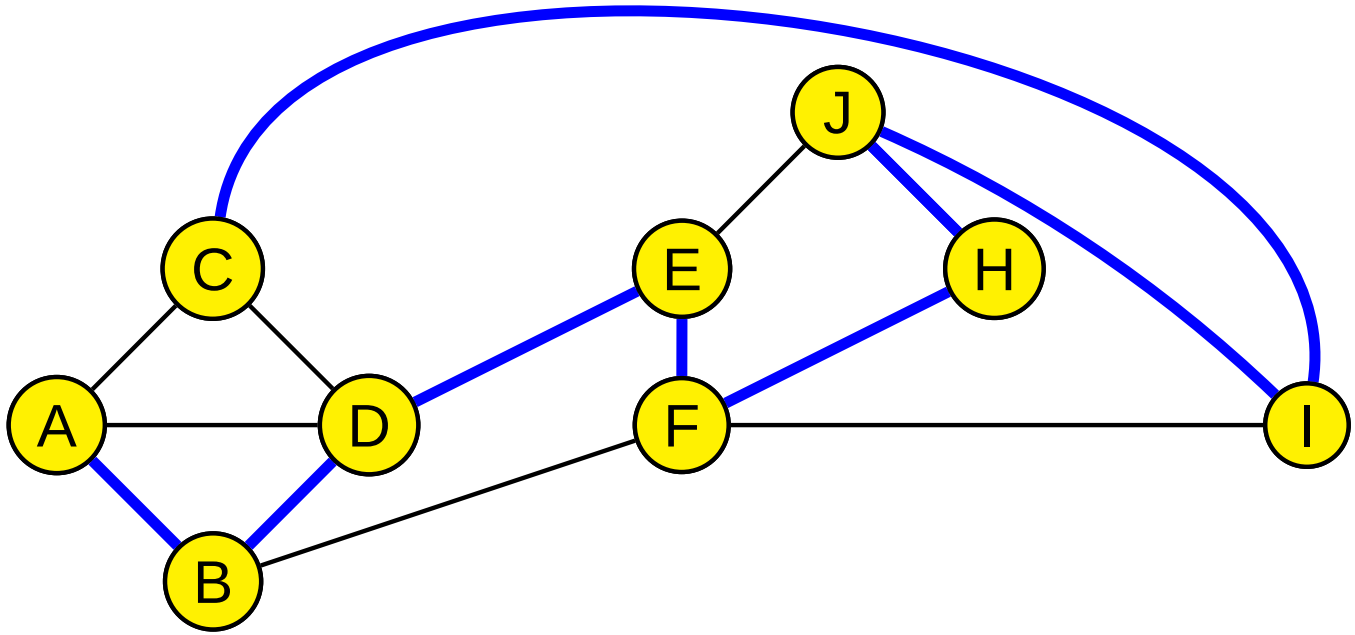
- n^{100} n'est pas une complexité "acceptable".
- Certains algorithmes polynomiaux ont été très durs... à trouver (et à comprendre).

Trouver un témoin... et vite

Les problèmes “difficiles” que l’on a vu sont très particuliers :
on peut vérifier facilement qu’un candidat est une solution !

Vérification d’une solution

Si on a $G = (V, E)$ et un circuit $f : \{1, .., n\} \longrightarrow V$, déterminer si f décrit un circuit hamiltonien est facile...



Trouver un témoin... et vite

Situation fréquente et naturelle...

k-colorabilité

Soient $G = (V, E)$ et un coloriage des sommets $c : V \longrightarrow \{1, ..., k\}$, déterminer si c décrit un “bon” k-coloriage se fait en temps polynomial.

Pareil pour coloration d’arêtes, circuit eulérien, etc.

Non primalité

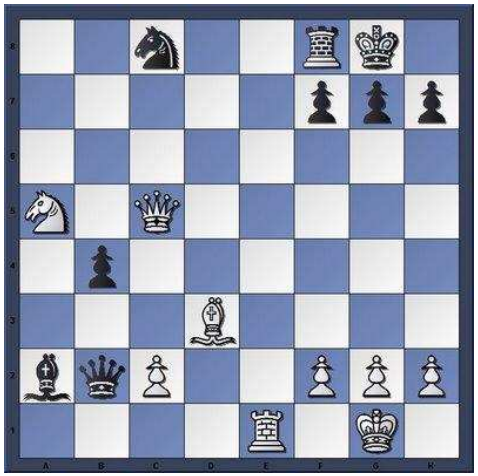
Soient trois entiers n, p, q, déterminer si $p \neq 1, n$ et $p \times q = n$ se fait en temps polynomial...

Trouver un témoin... et vite

Remarque

Tous les problèmes "difficiles" (ici, exponentiels) n'ont pas forcément cette propriété

Par exemple, déterminer l'existence d'une stratégie gagnante dans certains jeux classiques ou leur généralisation (les echecs, le Go, ...).



Le temps polynomial non déterministe **NP**

Vérification en temps polynomial

Un problème B est vérifiable en temps polynomial si pour toute instance x :

- si $x \in B$, il existe un témoin y (de taille polynomiale en la taille de x) permettant de décider en temps polynomial que $x \in B$.
- si $x \notin B$, il n'existe pas de tel témoin

Le temps polynomial non déterministe **NP**

Un problème de décision B est dans **NP** s'il est vérifiable en temps polynomial

Formalisé par Cook (1971) et Levin (1973).
Notions similaires : Edmonds (1966) mais aussi Gödel (1956 - Lettre à Von Neumann).

$\mathbf{P} = \mathbf{NP}$: un problème de témoin

$\mathbf{P} = \mathbf{NP}$?

Les problèmes vérifiables en temps polynomial sont-ils aussi décidables en temps polynomial ?
Autrement dit, existe-t'il des problèmes dont les solutions sont faciles à vérifier mais sont dures à trouver ?

Montrer $\mathbf{P} = \mathbf{NP}$: montrer que tous les problèmes de **NP** sont dans **P**.

Réductibilité : un seul pour les représenter tous

Certains problèmes dans **NP** sont représentatifs de la difficulté de la classe. Ils sont dits **NP**-complets

Outil principal : algorithme polynomial pour "réduire" un problème à un autre, une réduction.

Cook (1971), Levin (1973)

Le problème *SAT* (satisfaisabilité d'une formule propositionnelle) est **NP**-complet

Depuis lors, plusieurs centaines de problèmes *NP*-complets prouvés (depuis travail de Karp (1972))...

Un exemple de réduction

Dans les graphes

recouvrement par sommets minimal

sous-ensemble S de taille minimale tel que toute arête a un de ses sommets dans S.

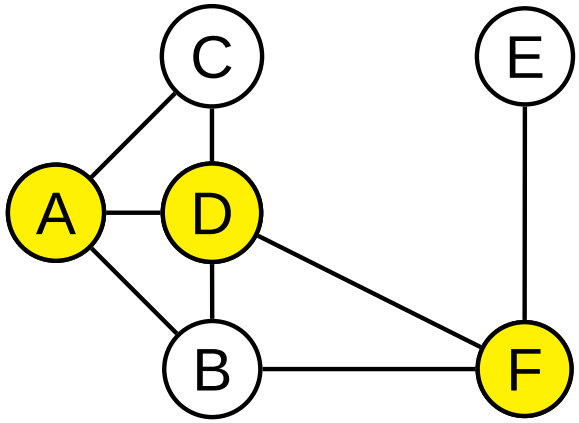
Clique maximale

sous-ensemble T de taille maximale tel que toute paire de sommets de T est liée par une arête.

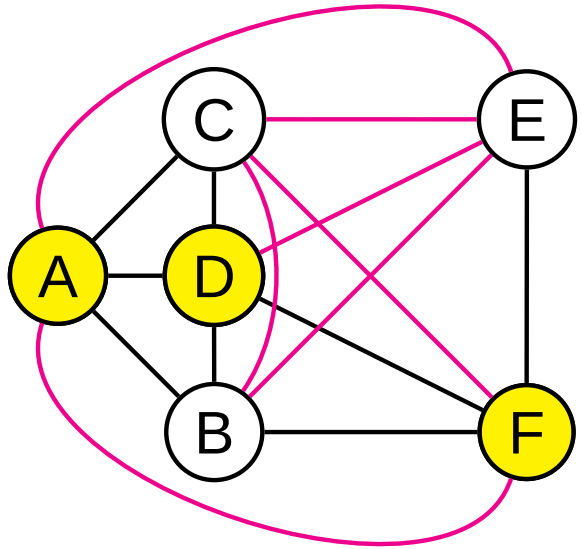
À partir d'un graphe G, construire en temps polynomial un graphe G' tel que

G a un recouvrement de taille minimale h ssi G' a une clique de taille maximale k.

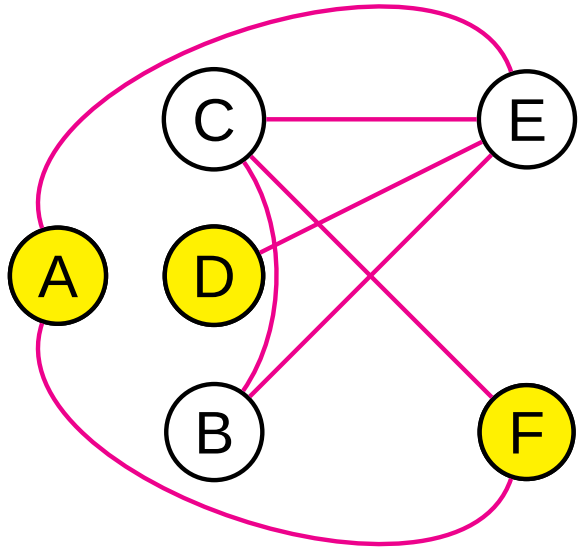
Un exemple de réduction



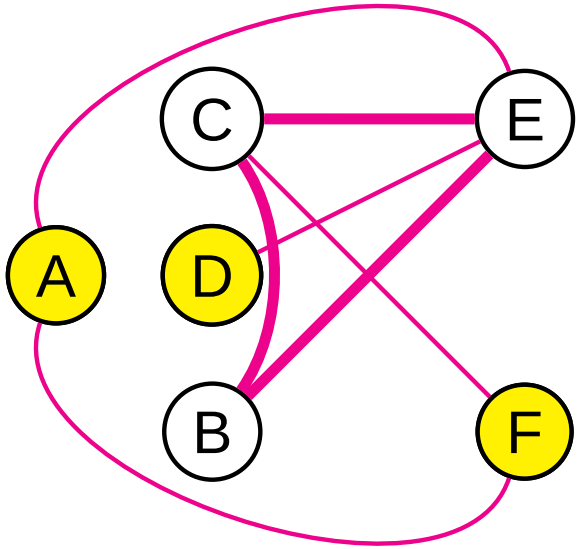
Un exemple de réduction



Un exemple de réduction



Un exemple de réduction



Réduction

Pour trouver un recouvrement minimal dans G, on cherche une clique maximale dans un autre graphe G', "facilement" constructible.

Réductibilité

Dans l'exemple précédent, trouver un recouvrement minimal se réduit à trouver une clique maximale.
Donc trouver une clique maximale est au moins aussi dur que trouver un recouvrement minimal.

un problème est **NP**-complet

si tout problème de **NP** se réduit à lui.

Conséquence

Trouver un algorithme polynomial pour un problème **NP**-complet donnerait automatiquement un algorithme polynomial pour tous les problèmes de **NP**
A priori, plein de façon d'attaquer la conjecture (chacun peut choisir son problème préféré...).

Quelques remarques sur la conjecture

Arguments pour $P \neq NP$:

- Pragmatique : beaucoup de gens d'horizons très différents ont cherché un algorithme polynomial pour un problème **NP**-complet...

Quelques approches pour la résolution pratique

1. Heuristique : toutes les instances d'un problème sont loin d'être "uniformément" difficiles
2. Approche probabiliste : trouver une solution avec une bonne probabilité
3. Approximation : trouver une solution proche de l'optimum
4. Approximation probabiliste : trouver avec une bonne probabilité une solution proche de l'optimum
5. Famille d'instances facile : Déterminer des familles entières d'instances pour lesquelles "**P** $=$ **NP**"

Quelques approches

1. Les méthodes actuelles de diagonalisation ne semblent pas convenir.
2. Les preuves ne passeront pas la relativisation. On sait qu'il existe deux ensembles A et B tels que :

$$\mathbf{P}^A = \mathbf{NP}^A \text{ et } \mathbf{P}^B \neq \mathbf{NP}^B$$

3. Approches combinatoires et probabilistes. Bons résultats avec de petites classes de complexité et caractérisations très importantes de **NP** (PCP).
4. Tentative de relier à d'autres questions, notamment l'hypothèse de Riemann.