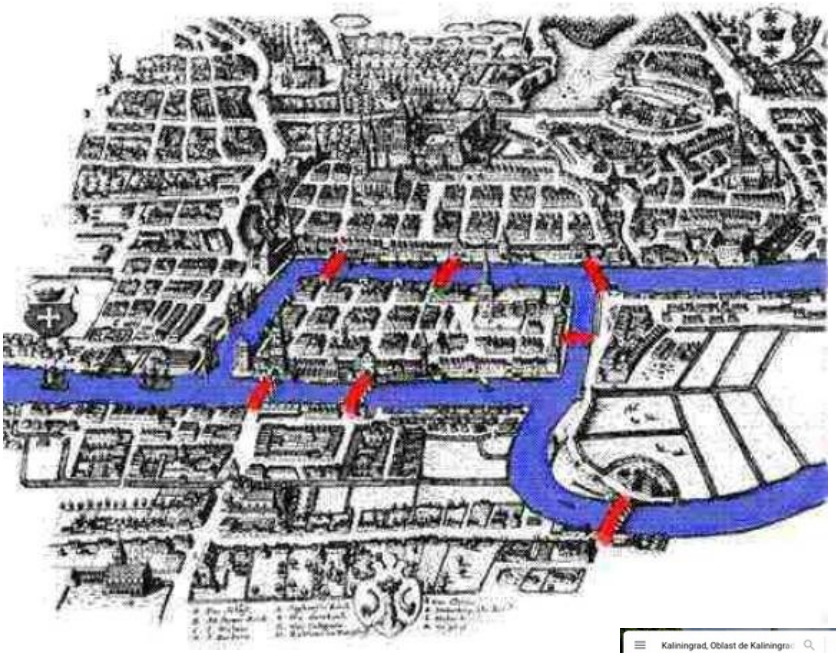
Université de Paris

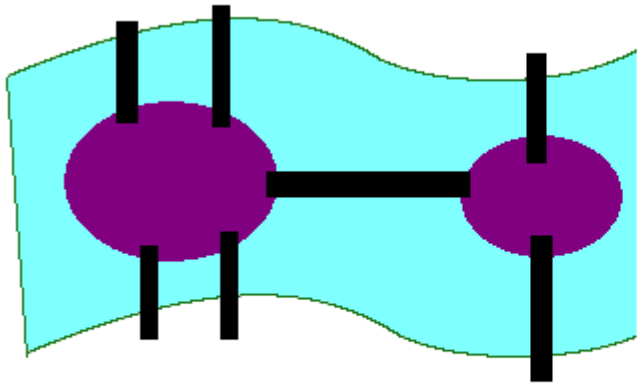
Algorithmie Avancée

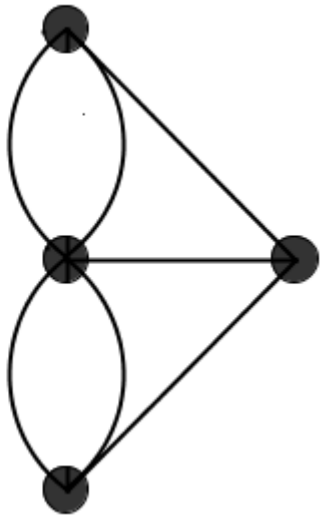
Mise en Contexte / Mise en Oeuvre

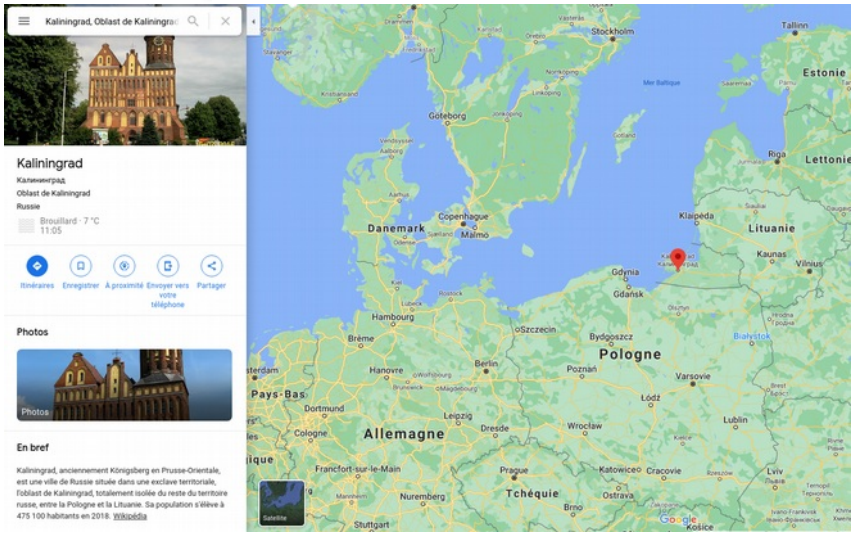
Année 2023-2024 par Prof. Nicolas Loménie
Sur la base du cours de Prof. Etienne Birmelé (2016-2020)

Mise en Contexte







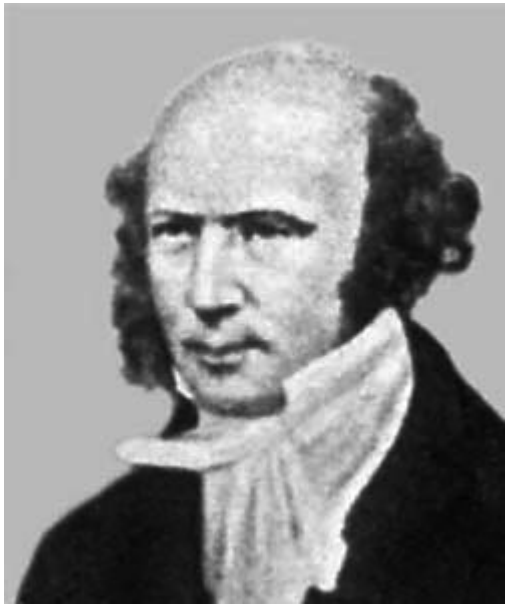


http://www.bibmath.net/dico/index.php?action=affiche&quoi=./p/pont.html

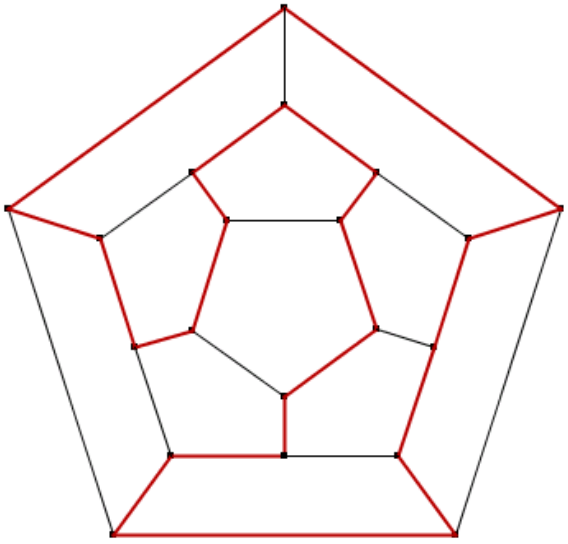
https://www.sciencedirect.com/science/article/abs/pii/S0030399214001637

https://www.youtube.com/watch?v=9Jpgzbrr-Kw

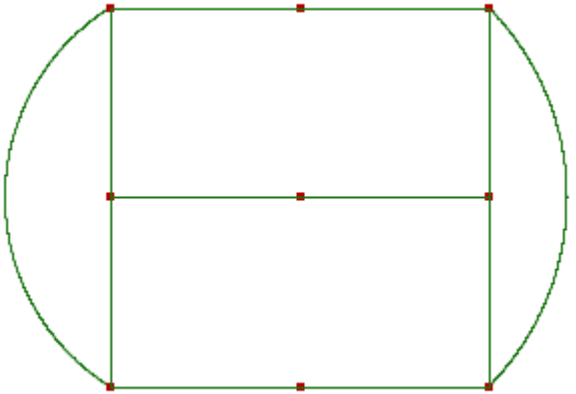
Mise en Contexte



William Rowan Hamilton (4 août 1805 [Dublin] - 2 septembre 1865 [Dublin])



Un graphe hamiltonien (un cycle hamiltonien est tracé en rouge)

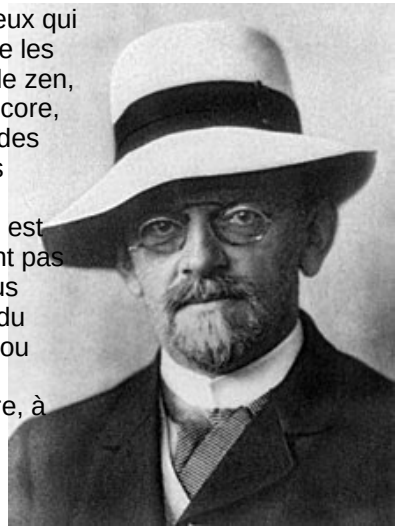


Un graphe qui ne possède pas de cycle hamiltonien

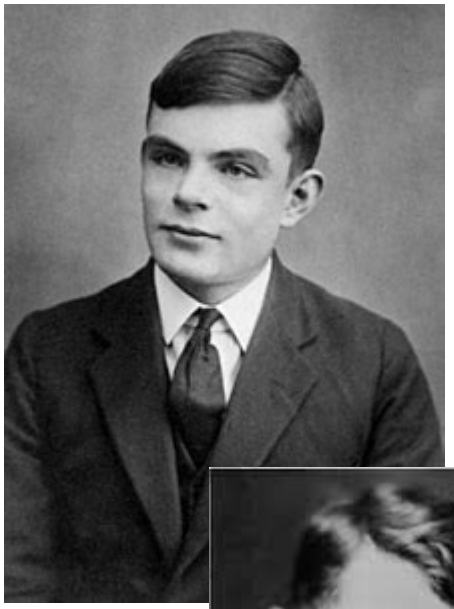
http://www.bibmath.net/dico/index.php?action=affiche&quoi=./h/hamiltonien.html

Mise en Contexte

L'ouvrage a de quoi surprendre: la logique, l'art et la musique réunis avec une adresse inégalée par le fils d'un prix nobel de physique... qui plus est, l'ouvrage obtient le prix pulizer! L'ouvrage est intelligent, il pousse à la réflexion, fait découvrir les problèmes logiques de la réflexivité - pour ceux qui ne les connaissaient pas - évoque les mathématiques, l'épistémologie, le zen, la théorie musicale, etc... Plus encore, chaque nouvelle lecture apporte des éléments nouveaux et stimule les neurones. On ressort du livre en n'ayant qu'une critique: mais qu'il est brillant! L'ouvrage n'est cependant pas accessible à tout le monde ou plus précisément, les connaissances du lecteur lui permettent d'aller plus ou moins vite dans la lecture et la compréhension de l'ouvrage. A lire, à relire et à réutiliser.



David Hilbert 1928

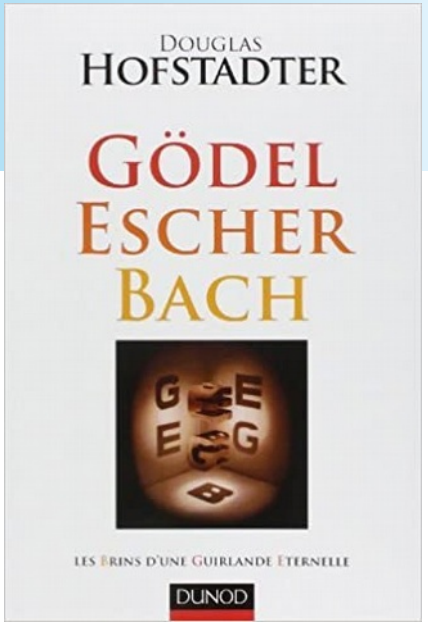


Kurt Gödel: théorème d'incomplétude

John Von Neumann : math et physique

Alan Turing
Alonzo Church 1930'

Logique formelle,
Complexité, NP-Complétude
Décidabilité etc.
Informatique théorique



https://fr.wikipedia.org/wiki/Probl%C3%A8me_P_%E2%89%9F_NP

Dans les années 1930, Princeton est un lieu propice aux échanges en logique car John von Neumann s'y trouve, ainsi que trois étudiants brillants de Church, Stephen Kleene, John Barkley Rosser et Alan Turing. Kurt Gödel, après plusieurs déplacements à l'Institute for Advanced Study entre 1933 et 1935, y donne plusieurs conférences sur son théorème d'incomplétude, et s'y installe définitivement vers 1940. L'Association for Symbolic Logic y naît ; il en est nommé le président en 1936. Church en sera l'un des éditeurs. Il édite 15 volumes entre 1936 et 1950. Il est également le rédacteur en chef de la partie Review du Journal of Symbolic Logic dans laquelle il apporte une relecture et une critique analytique des thèses qui lui sont soumises, une tâche qu'il accompli pour les 44 premiers volumes entre 1936 et 1979.

Théorie des Graphes 7

- AlgoAvanceeCoursPart1.pdf (fin)

Planche 120 à 126 (Fin Cycle Couvrant (Hamilton) – Chinese Postman Problem,

NP-Complétude)

https://www.jiblm.org/mahavier/discrete/html/section-5.html

https://fr.wikipedia.org/wiki/Graphe_hamiltonien

Mise en Oeuvre

La seconde propriété de la définition implique que s'il existe un algorithme polynomial pour résoudre un quelconque des problèmes NP-complets, alors tous les problèmes de la classe NP peuvent être résolus en temps polynomial. Trouver un algorithme polynomial pour un problème NP-complet ou prouver qu'il n'en existe pas permettrait de savoir si P = NP ou P ≠ NP, une question ouverte qui fait partie des problèmes non résolus en mathématiques les plus importants à ce jour.

En pratique, les informaticiens et les développeurs sont souvent confrontés à des problèmes NP-complets. Dans ce cas, savoir que le problème sur lequel on travaille est NP-complet est une indication du fait que le problème est difficile à résoudre, donc qu'il vaut mieux chercher des solutions approchées en utilisant des algorithmes d'approximation ou utiliser des heuristiques pour trouver des solutions exactes.

Le problème à 1 000 000 dollars

Le concept de NP-complétude a été introduit en 1971 par Stephen Cook dans une communication intitulée The complexity of theorem-proving procedures (La complexité des procédures de démonstration de problèmes), bien que le mot « NP-complet » n'apparaisse pas explicitement dans l'article. Lors de la conférence à laquelle il a été présenté, une discussion acharnée a eu lieu entre les chercheurs présents pour savoir si les problèmes NP-complets pouvaient être résolus en temps polynomial sur machine de Turing déterministe. John Hopcroft a finalement convaincu les participants que la question devait être remise à plus tard, personne n'ayant réussi à démontrer ou infirmer le résultat.

La question de savoir si P = NP n'est toujours pas résolue. Elle fait partie des problèmes du prix du millénaire, un ensemble de sept problèmes pour lesquels l'Institut de mathématiques Clay offre un prix d'un million de dollars. Il « suffirait » de trouver un seul problème NP qui soit à la fois NP-complet et P pour démontrer cette hypothèse, ou d'exhiber un seul problème NP qui ne soit pas dans P pour démontrer sa négation.

Le résultat de l'article de Cook, démontré de manière indépendante par Leonid Levin en URSS, est maintenant connu sous le nom de théorème de Cook-Levin. Ce théorème affirme qu'il existe un problème NP-complet. Cook a choisi le problème SAT et Levin un problème de pavage. En 1972, Richard Karp a prouvé la NP-complétude de plusieurs autres problèmes fondamentaux en informatique et très disparates, connus comme la liste des 21 problèmes NP-complets de Karp. Depuis, on a démontré la NP-complétude de milliers d'autres problèmes.

https://fr.wikipedia.org/wiki/Probl%C3%A8me_NP-complet
https://www.lemonde.fr/blog/binaire/2015/04/07/une-complete-incompletude/
https://fr.wikipedia.org/wiki/Probl%C3%A8mes_du_prix_du_mill%C3%A9naire

Mise en Oeuvre

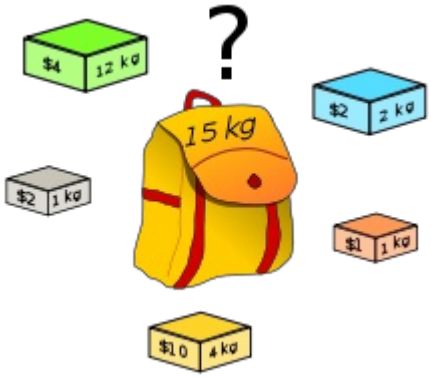
$$P \subseteq NP \subseteq NP\text{-}Complet \subseteq NP\text{-}Difficile$$

= ?

1 000 000 dollars

Vérification d'une Solution en temps Polynomial
Certificat polynomial

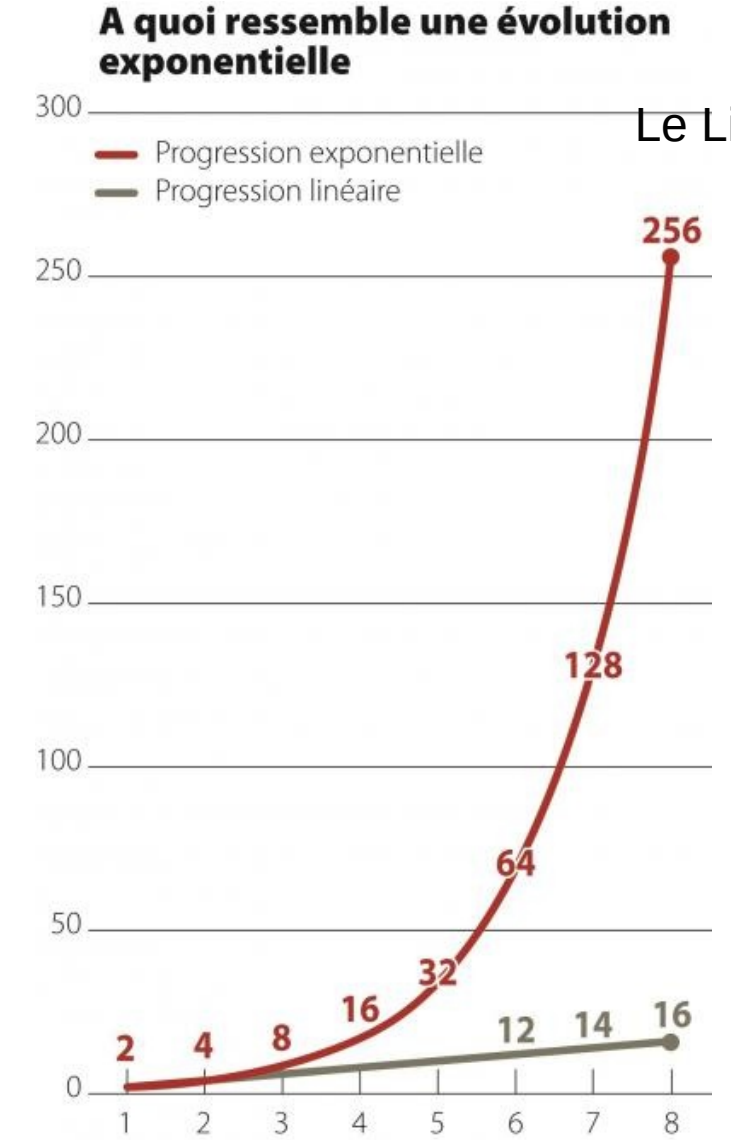
Pb du Voyageur de Commerce
Ou
Du Sac à dos



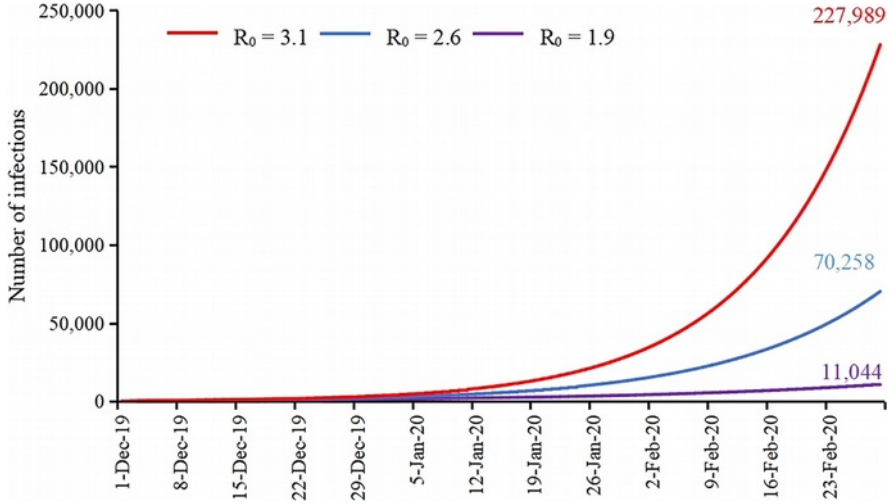
Pour tous les problèmes connus NP-complets, **on ne peut pas trouver une solution en temps polynomial** mais seulement en temps d'exécution exponentiel en la taille des données dans le pire des cas, ce qui est inexploitable en pratique même pour des instances de taille modérée

https://en.wikipedia.org/wiki/Knapsack_problem

Mise en Oeuvre



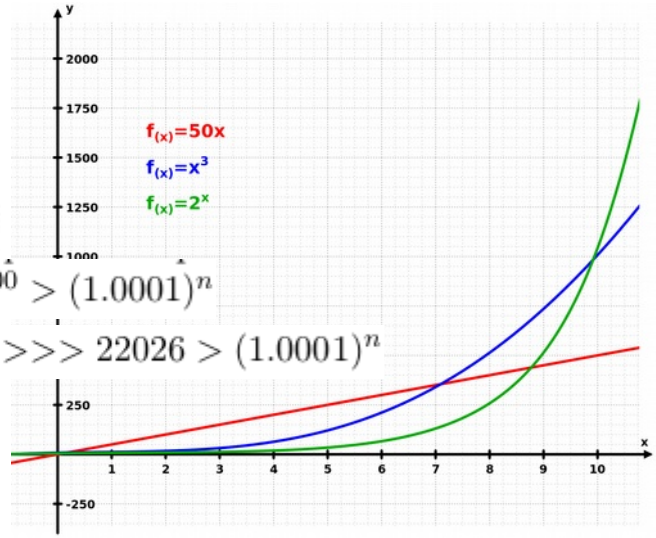
Le Livret A :-)



n^{1000} vs $(1.0001)^n$

. Pour $n = 10^9$, $n^{1000} > (1.0001)^n$

$n = 10^6$, on a 10^{6000} >>> $22026 > (1.0001)^n$



En général, les polynômes sont de degré inférieur à 10, et le plus souvent le degré ne dépasse pas 3 ou 4. Donc en général la solution polynomiale est bien meilleure que la solution exponentielle

https://fr.wikipedia.org/wiki/Croissance_exponentielle

https://couleur-science.eu/?d=d3fdf5--la-fonction-exponentielle

Mise en Oeuvre

Théorème de Cook : SAT est NP-Complet

En informatique théorique, le problème SAT ou problème de satisfaisabilité booléenne est le problème de décision, qui, étant donné une formule de logique propositionnelle, détermine s'il existe une assignation des variables propositionnelles qui rend la formule vraie.

Définition 10 (SAT). *Soit F une formule booléenne à n variables $x_1, x_2, ..., x_n$ et p clauses C_i : $F = C_1 \wedge C_2 \wedge ... \wedge C_p$ où $C_i = x^*_{i_1} \vee x^*_{i_2} \vee ... \vee x^*_{i_{f(i)}}$ et $x^* = x$ ou $\overline{x}$.*

Existe-t-il une instanciation des variables telle que F soit vraie ($\Leftrightarrow C_i$ vraie pour tout i) ?

Mise en Oeuvre

Théorème de Cook : SAT est NP-Complet

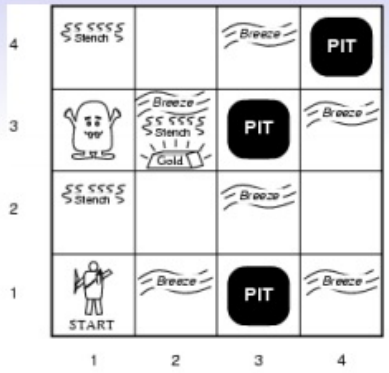
Définition 10 (SAT). *Soit F une formule booléenne à n variables $x_1, x_2, ..., x_n$ et p clauses C_i :* $F = C_1 \wedge C_2 \wedge ... \wedge C_p$ *où* $C_i = x^*_{i_1} \vee x^*_{i_2} \vee ... \vee x^*_{i_{f(i)}}$ *et* $x^* = x$ *ou* $\overline{x}$.

Existe-t-il une instanciation des variables telle que F soit vraie ($\Leftrightarrow C_i$ vraie pour tout i) ?

Agents logiques

Le monde du Wumpus

- Environnement
  - Agent commence en case [1,1]
  - Cases adjacentes au Wumpus sentent mauvais
  - Brise dans les cases adjacentes aux puits
  - Lueur dans la cases contenant de l'or
  - Tirer tue le Wumpus s'il est en face
  - On ne peut tirer qu'une fois
  - S'il est tué, le Wumpus crie
  - Choc si l'agent se heurte à un mur
  - Saisir l'or si même case que l'agent
- Capteurs : odeur, brise, lueur, choc, cri
- Percepts : liste de 5 symboles
  Ex : [odeur, brise, rien, rien, rien]
- Actions : tourne gauche, tourne droite, avance, attrappe, tire



- Mesures de performance :
  - or : +1000;
  - mort : -1000;
  - action : -1;
  - utiliser la flèche : -10

- $P_{i,j}$ vrai s'il y a un puits en $[i, j]$
- $B_{i,j}$ vrai s'il y a une brise en $[i, j]$
- Base de connaissances :
  - $R_1 : \neg P_{1,1}$
  - Brise ssi puits dans une case adjacente :
    $R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
    $R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
  - $R_4 : \neg B_{1,1}$
  - $R_5 : B_{2,1}$
- BC : $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

http://helios.mi.parisdescartes.fr/~bouzy/Doc/IAL3/06_IA_logique_BB.pdf

https://laurent-audibert.developpez.com/Cours-Logique/?page=logique-propositions

Mise en Oeuvre

Soit un problème **Q** qui est NP-complet par exemple **SAT** (théorème de Cook)
Montrer que **Prob** est NP-Complet revient à trouver une **réduction polynomiale** de **Prob** à partir de Q

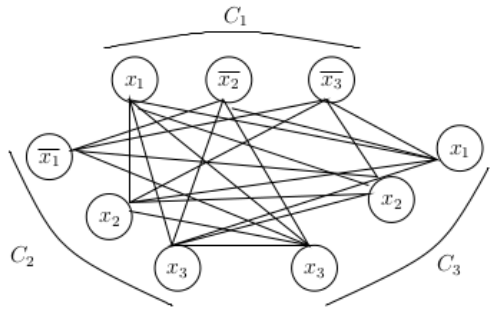
Prob = REDUCTION(**Q**) && **Q** NP-Complet ⟶ **Prob** NP-Complet

3-SAT = REDUCTION(**SAT**) ⟶ **3-SAT** NP-Complet

3-SAT toutes les clauses du problème **SAT** ont exactement 3 termes

Mais 2-SAT peut être résolu en temps polynomial :-)

CLIQUE = REDUCTION(**3-SAT**) ⟶ **CLIQUE** NP-Complet



Définition 11 (CLIQUE). *Soit un graphe $G = (V, E)$, et un entier k tel que $1 \leq k \leq |V|$.*
Existe-t-il une clique de taille k (un sous graphe complet de k sommets) ?
Taille d'une instance : $|V| + |E|$ ou $|V|$ (car $|E| \leq |V|^2$).

https://fr.wikipedia.org/wiki/Probl%C3%A8me_SAT

https://en.wikipedia.org/wiki/Reduction_(complexity)

Mise en Oeuvre

VERTEX_COVER = REDUCTION(**CLIQUE**) ⟶ **VERTEX_COVER** NP-Complet

Définition 12 (VERTEX-COVER). *Soit un graphe $G = (V, E)$, et un entier k : $1 \leq k \leq |V|$ Existe-t-il k sommets $v_{i_1}, ..., v_{i_k}$ tels que $\forall e \in E$, e est adjacente à l'un des v_{i_j} ?*

CH = REDUCTION(**3-SAT**) ⟶ **CH** NP-Complet

Définition 13 (CH). *Soit un graphe $G = (V, E)$. Existe-t-il un cycle hamiltonien dans G, c'est à dire un chemin qui visite tous les sommets une fois et une seule et revient à son point de départ ?*

COLOR = REDUCTION(**3-SAT**) ⟶ **COLOR** NP-Complet

Définition 14 (COLOR). *Soit $G = (V, E)$ un graphe et k une borne $(1 \leq k \leq |V|)$. Peut-on colorier les sommets de G avec k couleurs ? Le problème est d'associer à chaque sommet $v \in V$ un nombre (sa couleur) $color(v)$ compris entre 1 et k, de telle sorte que deux sommets adjacents ne reçoivent pas la même couleur :*

$$(u, v) \in E \Rightarrow color(u) \neq color(v)$$

Mise en Oeuvre

3-COLOR = REDUCTION(**3-SAT**) ⟶ **3-COLOR** NP-Complet

Définition 15 (3-COLOR). *Soit $G = (V, E)$ un graphe. Peut on colorier les sommets de G avec 3 couleurs ?*

Mais 2-COLOR est dans **P** : graphe bi-parti via DFS (cycles tous pairs) etc.

Mise en Oeuvre

Chevaliers de la table ronde
Étant donnés n chevaliers, et connaissant toutes les paires de féroces ennemis parmi eux, est-il possible de les placer autour d'une table circulaire de telle sorte qu'aucune paire de féroces ennemis ne soit côte à côte ?

Etant donné un graphe G = (V, E), et un entier K > 0, existe-t-il deux cliques disjointes de taille K dans G ? On rappelle qu'une clique est un sous-ensemble de sommets tous adjacents deux à deux.

Soient G = (V, E) un graphe orienté (on distingue l'arc (u,v) de l'arc (v,u)), deux sommets de ce graphe s, t ∈ V et une liste P = $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ de paires de sommets de G. Existe-t-il un chemin orienté de s vers t dans G qui contient au plus un sommet de chaque paire de la liste P ?

Soient G = (V, E) un graphe dont tous les sommets sont de degré pair, et
k ≥ 1 un entier. Existe-t-il un ensemble de sommets de G couvrant toutes les arêtes et de taille inférieure ou égale à k ?

Mise en Oeuvre

Soient n = 2p un entier pair, et n entiers strictement positifs $a_1, a_2, \ldots, a_n$. Existe-t-il une P partition de {1, 2, . . . n} en deux ensembles I et I' de même cardinal p et tels P que $\sum_{i \in I} a_i = \sum_{i \in I'} a_i$?

Soient G = (V, E) un graphe dont tous les sommets sont de même degré, et k ≥ 1 un entier. Existe-t-il une clique de taille supérieure ou égale à k ?

Étant donnés un graphe G = (V, E) et un entier K ≥ 3, déterminer si G contient une roue de taille K, i.e. un ensemble de K + 1 sommets w, $v_1, v_2, \ldots, v_K$ tels que $(v_i, v_{i+1}) \in E$ pour 1 ≤ i < K, $(v_K, v_1) \in E$ et $(v_i, w) \in E$ pour 1 ≤ i ≤ K (w est le centre de la roue).

Étant donnés un graphe G = (V, E) et un entier K ≥ 3, déterminer si G contient un dominateur de cardinal K, i.e. un sous-ensemble D ⊂ V de cardinal K tel que pour tout sommet u ∈ V \ D, il existe u ∈ D avec (u, v) ∈ E

https://www2.math.upenn.edu/~wilf/AlgoComp.pdf

https://www2.math.upenn.edu/~wilf/

Mise en Oeuvre

Quelques bibliothèques Java accessibles pour manipuler les graphes ?

Pas d'implémentation native a priori à ce jour mais de nombreuses librairies disponibles.

JGraphT est l'une des bibliothèques les plus populaires en Java pour la structure de données graphiques. Il permet la création d'un graphe simple, d'un graphe orienté, d'un graphe pondéré, entre autres. De plus, il offre de nombreux algorithmes possibles sur la structure de données du graphe. Un de nos précédents tutoriels couvre la JGraphT de manière beaucoup plus détaillée .

Google Guava est un ensemble de bibliothèques Java offrant une gamme de fonctions, notamment la structure de données graphiques et ses algorithmes. Il prend en charge la création simple de Graph , ValueGraph et Network . Ceux-ci peuvent être définis comme Mutable ou Immutable .

Apache Commons est un projet Apache qui propose des composants Java réutilisables. Cela comprend Commons Graph, qui propose une boîte à outils permettant de créer et de gérer la structure des données graphiques. Cela fournit également des algorithmes de graphes communs pour agir sur la structure de données.

Java JUNG (Universal Network/Graph) est un framework Java qui fournit un langage extensible pour la modélisation, l'analyse et la visualisation de toutes les données pouvant être représentées sous forme de graphique. JUNG prend en charge un certain nombre d'algorithmes qui incluent des routines telles que la mise en cluster, la décomposition et l'optimisation.

Ces bibliothèques fournissent un certain nombre d'implémentations basées sur la structure de données de graphe. Il existe également des cadres plus puissants basés sur des graphes , tels que Apache Giraph , actuellement utilisé sur Facebook pour analyser le graphe formé par leurs utilisateurs https://tinkerpop.apache.org/ [Apache TinkerPop], couramment utilisé en plus des bases de données graphiques.

Et en C++ ? Python ? A vous de JOUER

https://www.codeflow.site/fr/article/java-graphs

https://algotree.org/algorithms/adjacency_list/graph_as_adjacency_list_java/

https://www.geeksforgeeks.org/implementing-generic-graph-in-java/ implementing HashMap

https://www.geeksforgeeks.org/graph-and-its-representations/