

Mardi 3 décembre

- Fondamentaux pour le calcul quantique
 - Qubit, définition
 - Qubit, manipulation
 - Plusieurs qubits.
-
- TP Exercices avec python, sans qiskit
 - TP Algorithmique (classique)
 - TP IBM Q experience



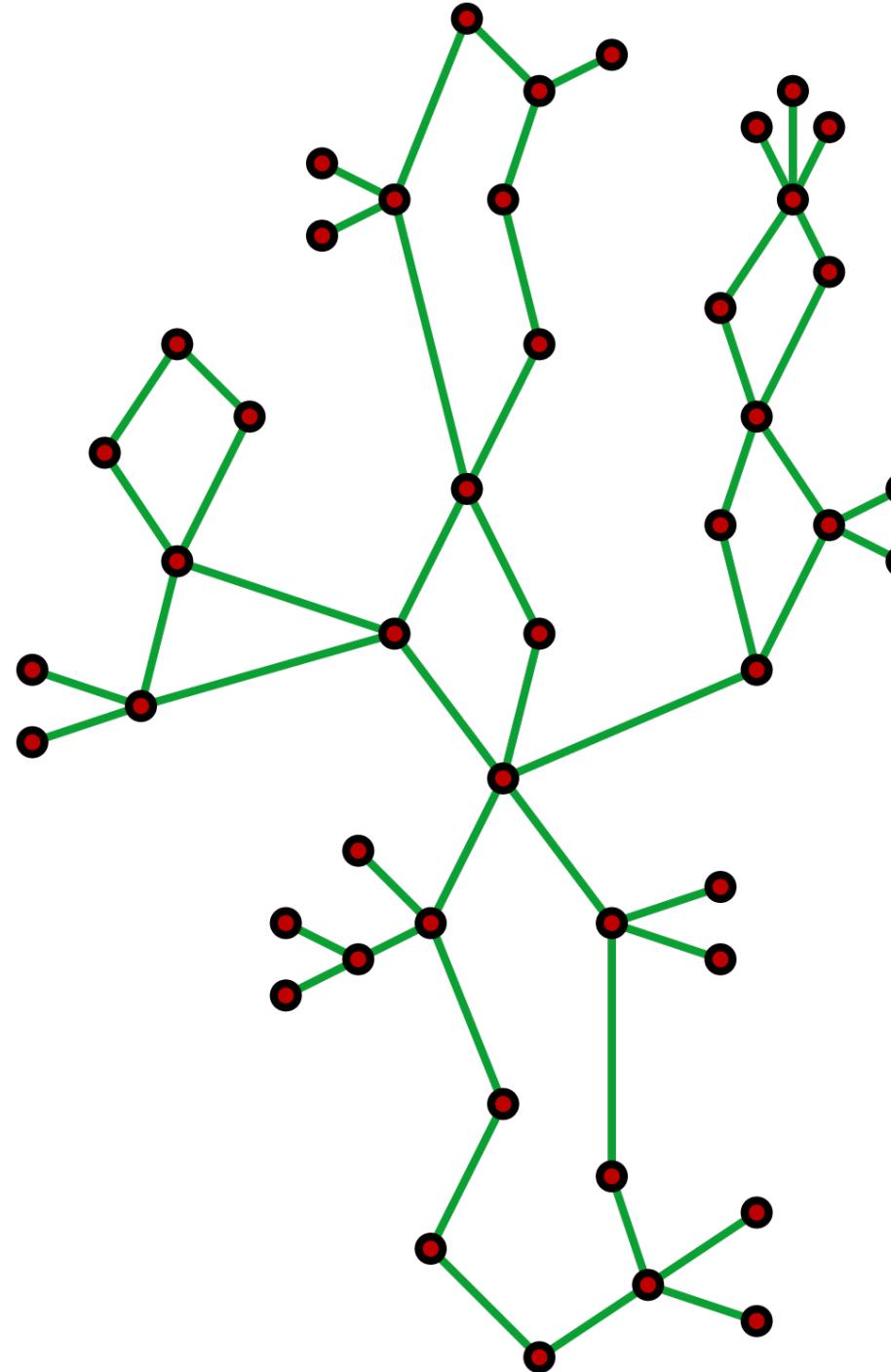
Fondamentaux pour le calcul quantique

IBM Client Center Montpellier

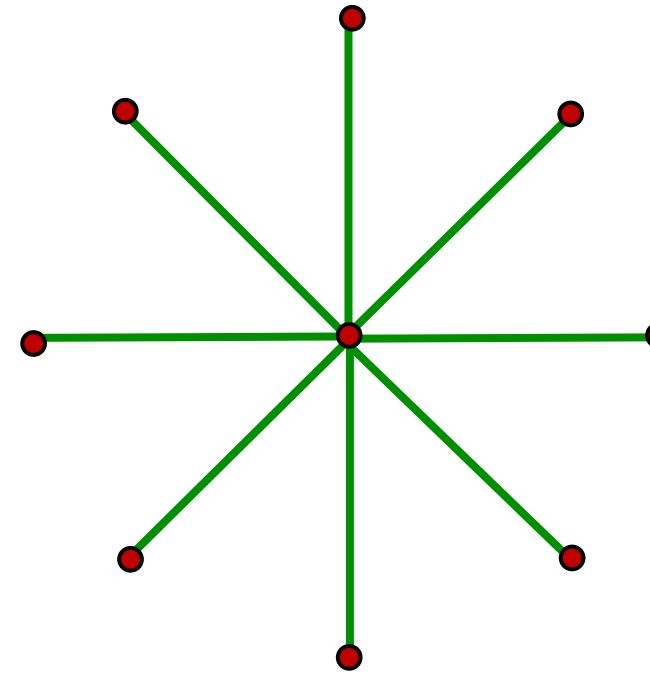
JM Torres | torresjm@fr.ibm.com | 2-4 décembre 2019

un graphe

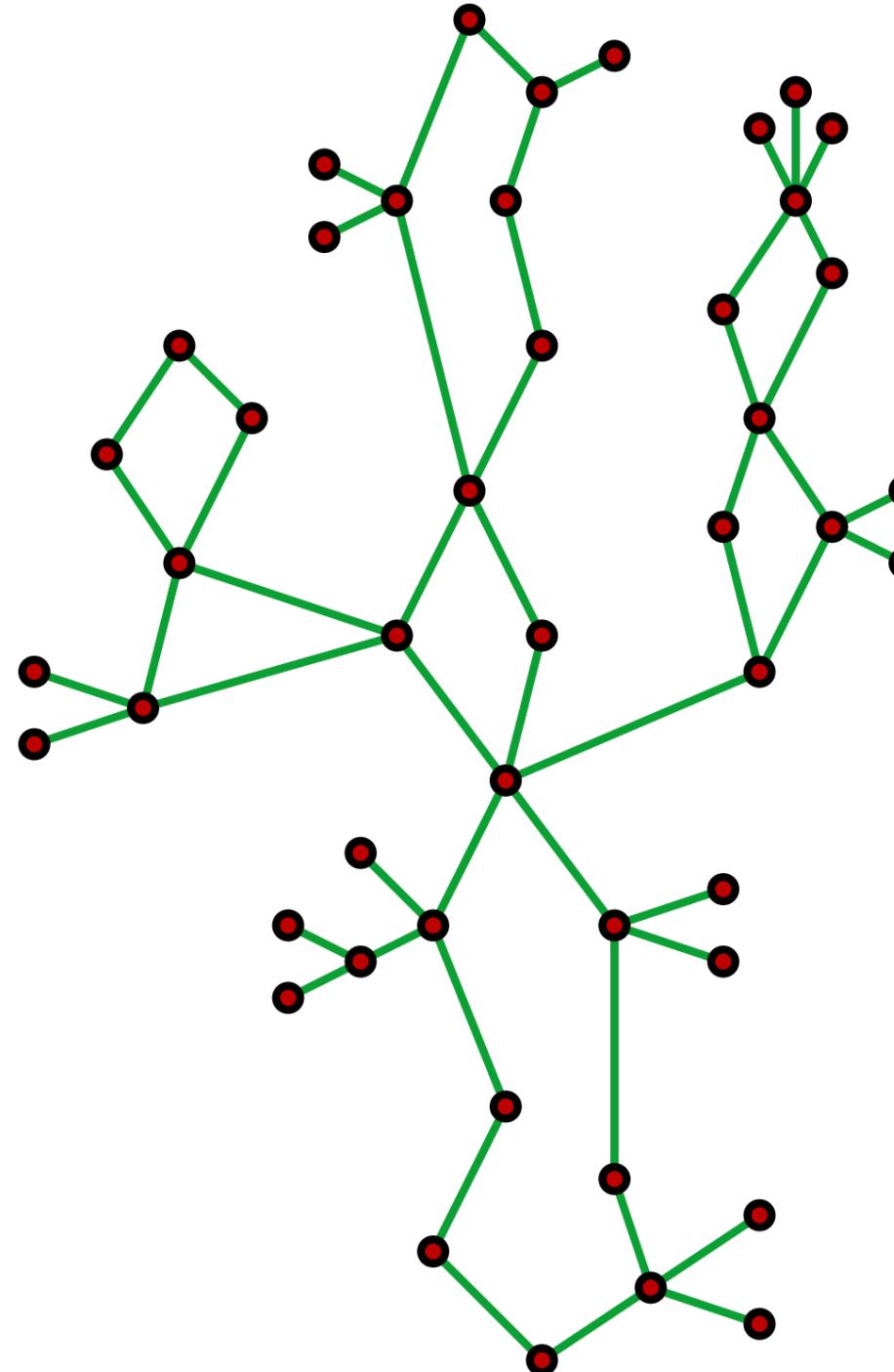
Comment construire un graphe planaire d'ordre n de diamètre minimal, et quel est ce diamètre ?



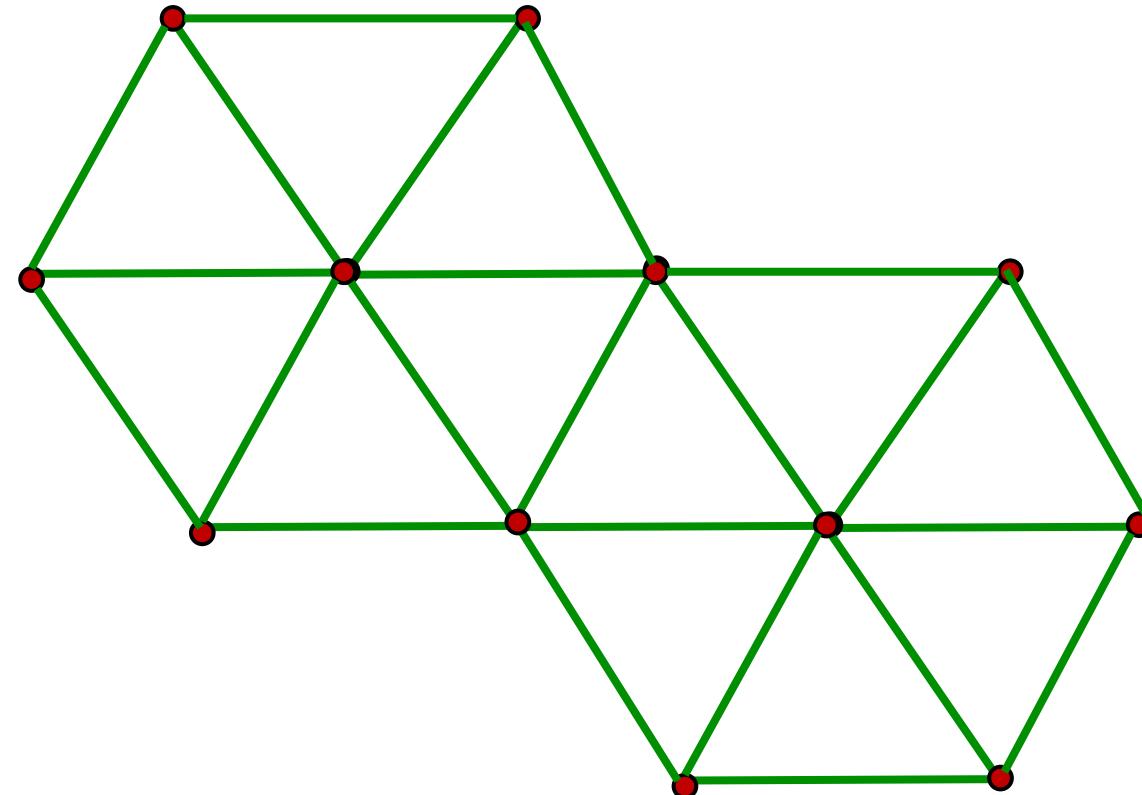
2



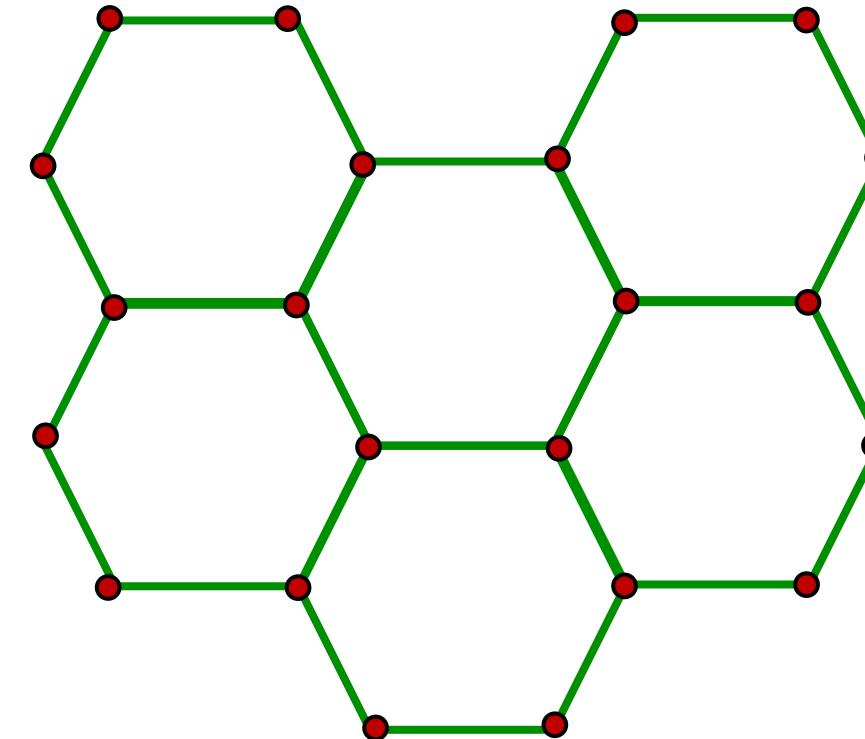
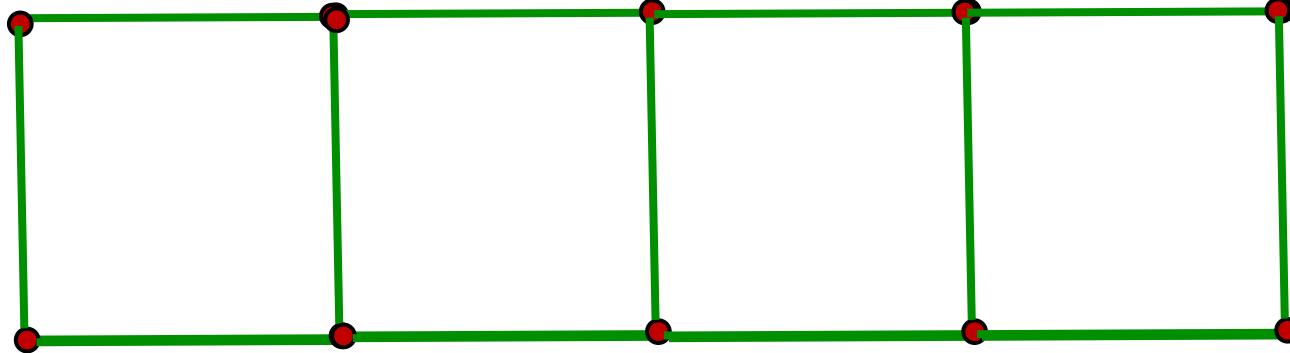
Comment construire
un graphe planaire
d'ordre n , dont
l'ordre des sommets
est inférieur ou égal
à 6, de diamètre
minimal, et quel est
ce diamètre ?



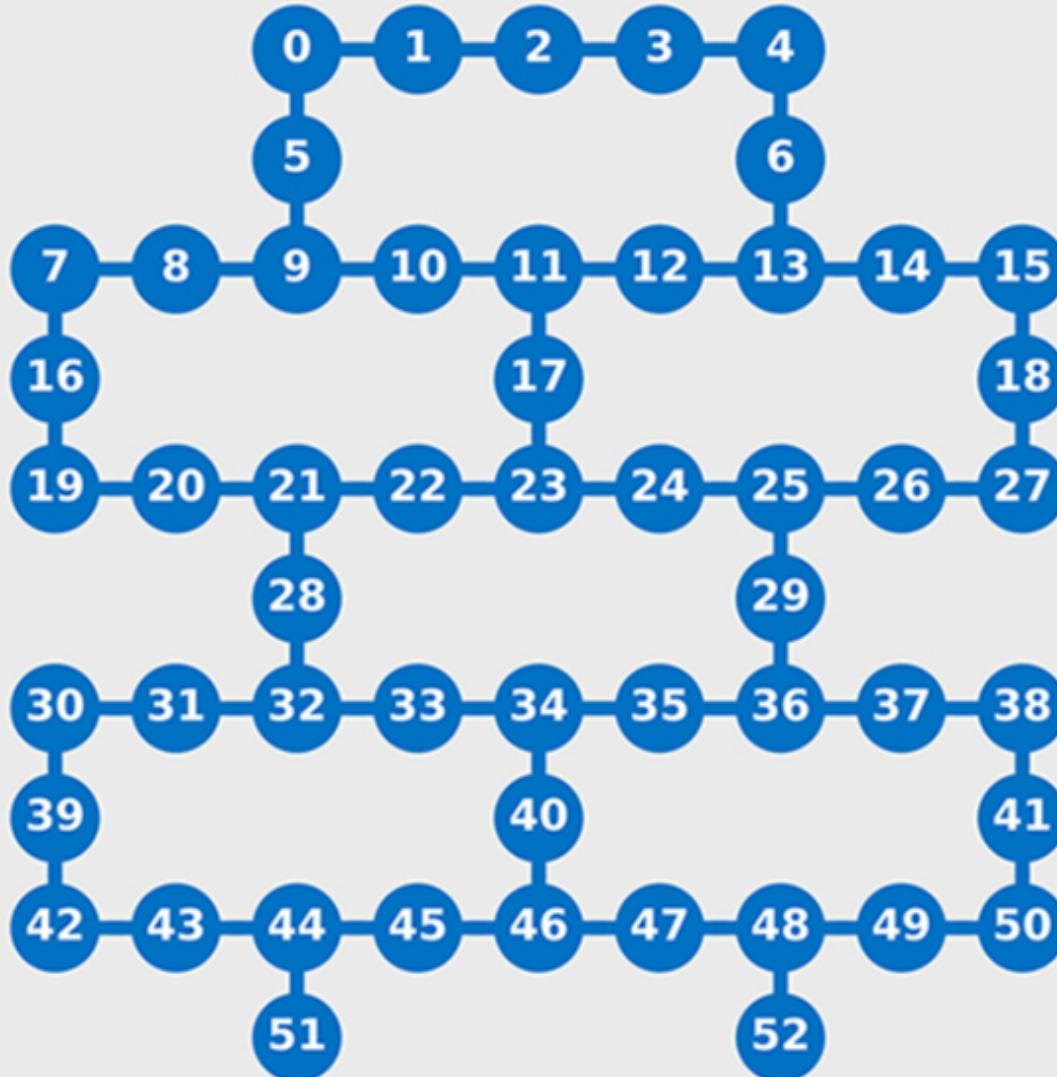
Comment construire
un graphe planaire
d'ordre n , dont
l'ordre des sommets
est inférieur ou égal
à 6, de diamètre
minimal, et quel est
ce diamètre ?



Comment construire
un graphe planaire
d'ordre n , dont
l'ordre des sommets
est inférieur ou égal
à 3, de diamètre
minimal, et quel est
ce diamètre ?



Machine IBM
53 qubit
("Rochester")



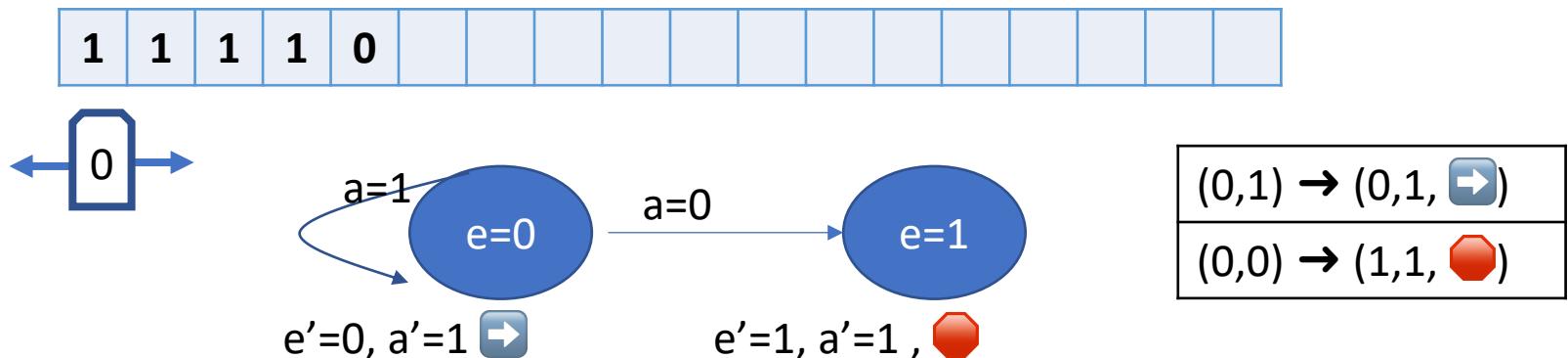
machine de turing, circuits

Machine de Turing



- un « ruban », infini
- une tête de lecture/écriture
- un ensemble d'états e_0, e_1, e_2, \dots
- un alphabet a_0, a_1, a_2, \dots
- une table de transitions, liste de:
(état courant, symbole sous la tête) \rightarrow (nouvel état,
nouveau symbole sous la tête
déplacement: $\rightarrow/\leftarrow/\text{stop}$)

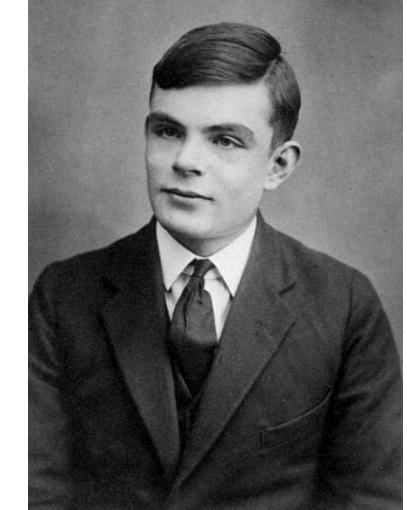
Par exemple, une machine qui incrémente un nombre n , écrit en unaire : $n \rightarrow n+1$



Thèse de Church-Turing

En 1936 Alan Turing établit qu'il existe des machines de Turing universelles (capable de simuler n'importe quelle machine de Turing)

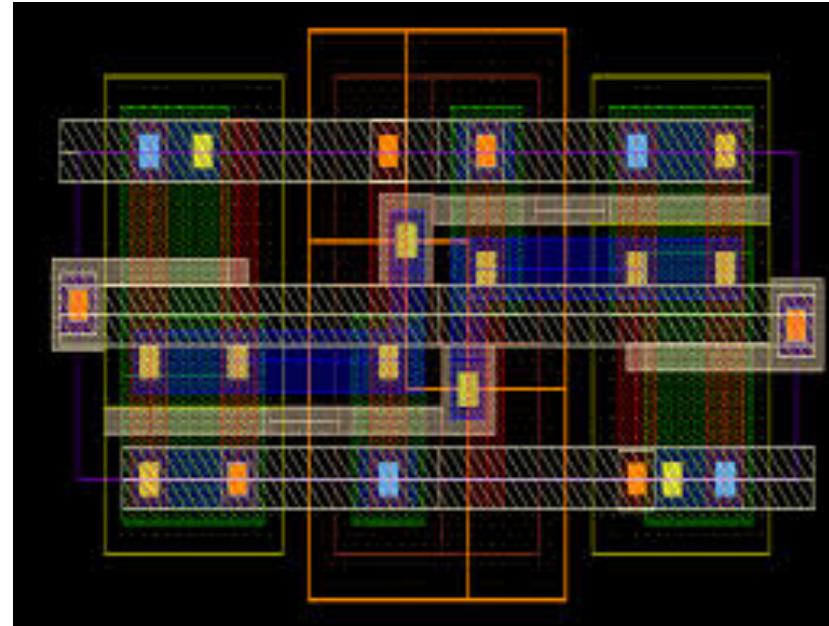
A la même période, la thèse de Church-Turing est énoncée : **tout calcul peut -être simulé par une machine de Turing.**



En 2002 Yurii Rogozin démontre l'existence d'une machine de Turing universelle contenant seulement 4 états, 9 symboles et 24 transitions.

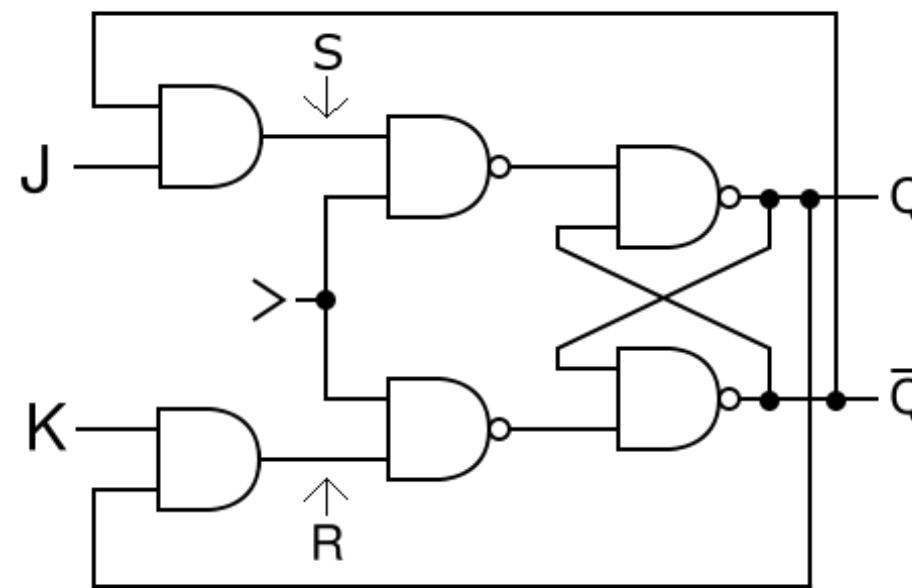
mais... les temps de calculs sont impraticables et le ruban peut être long

Le modèle à circuits



← ceci est un bit

Logique Booléenne



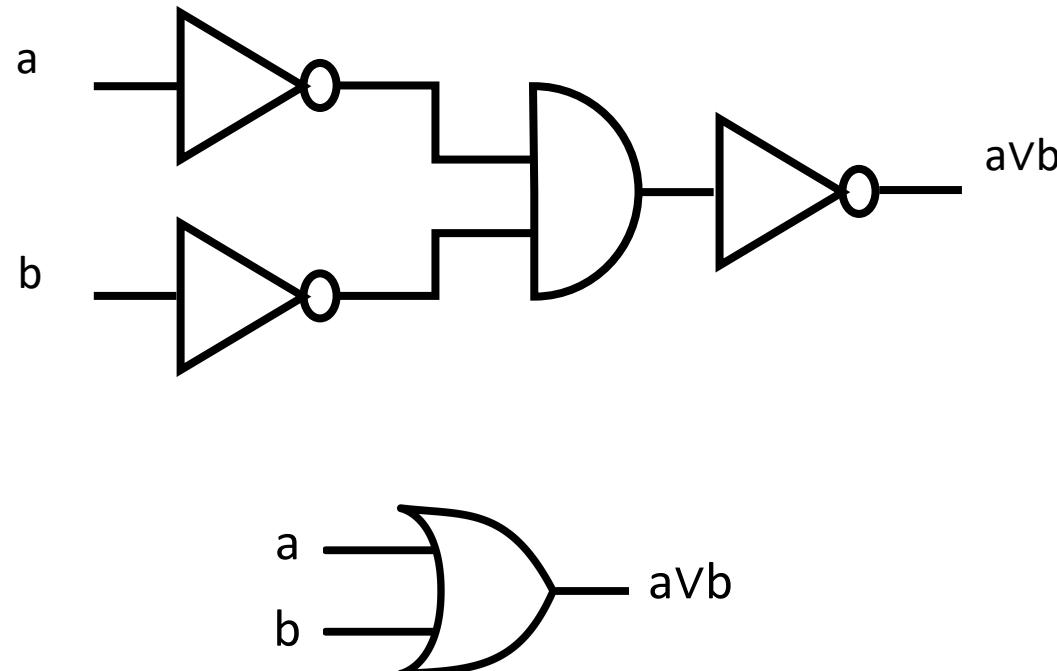
← ceci aussi

{NOT, AND} : ensemble de portes universel

universalité
des circuits

Théorème de Morgan: on peut faire un OR avec AND et NOT :

$$a \vee b = \neg(\neg a \wedge \neg b)$$



{NOT, AND} : ensemble de portes universel

Toute fonction logique peut être calculée avec des NOT et des AND (et donc des OR) :



universalité
des circuits

soit une fonction logique $f: n+1 \text{ bits} \rightarrow \{0,1\}$

on définit $f_0(X_1, X_2, \dots, X_n) = f(0, X_1, X_2, \dots, X_n)$

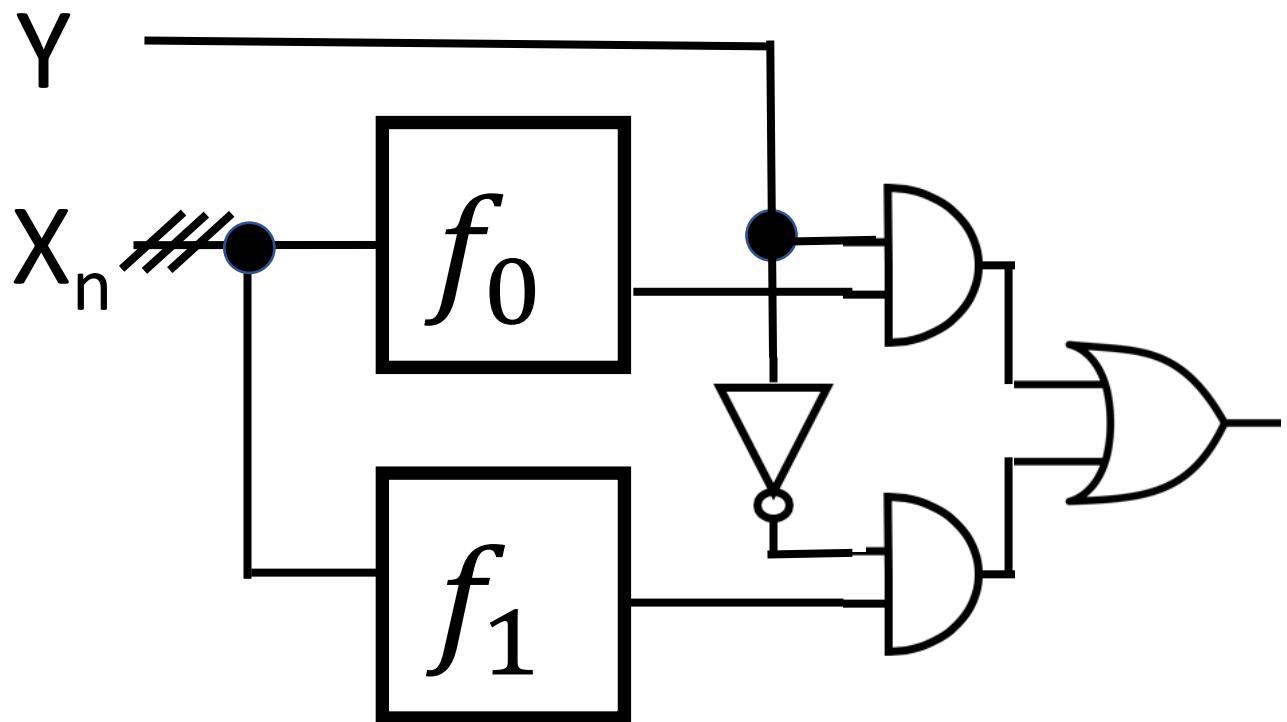
et $f_1(X_1, X_2, \dots, X_n) = f(1, X_1, X_2, \dots, X_n)$

Si f_0 & f_1 peuvent être calculées avec des NOT et des AND alors f aussi :

$$f(Y, X_1, X_2, \dots, X_n) = (\neg Y \wedge f_0) \vee (Y \wedge f_1)$$

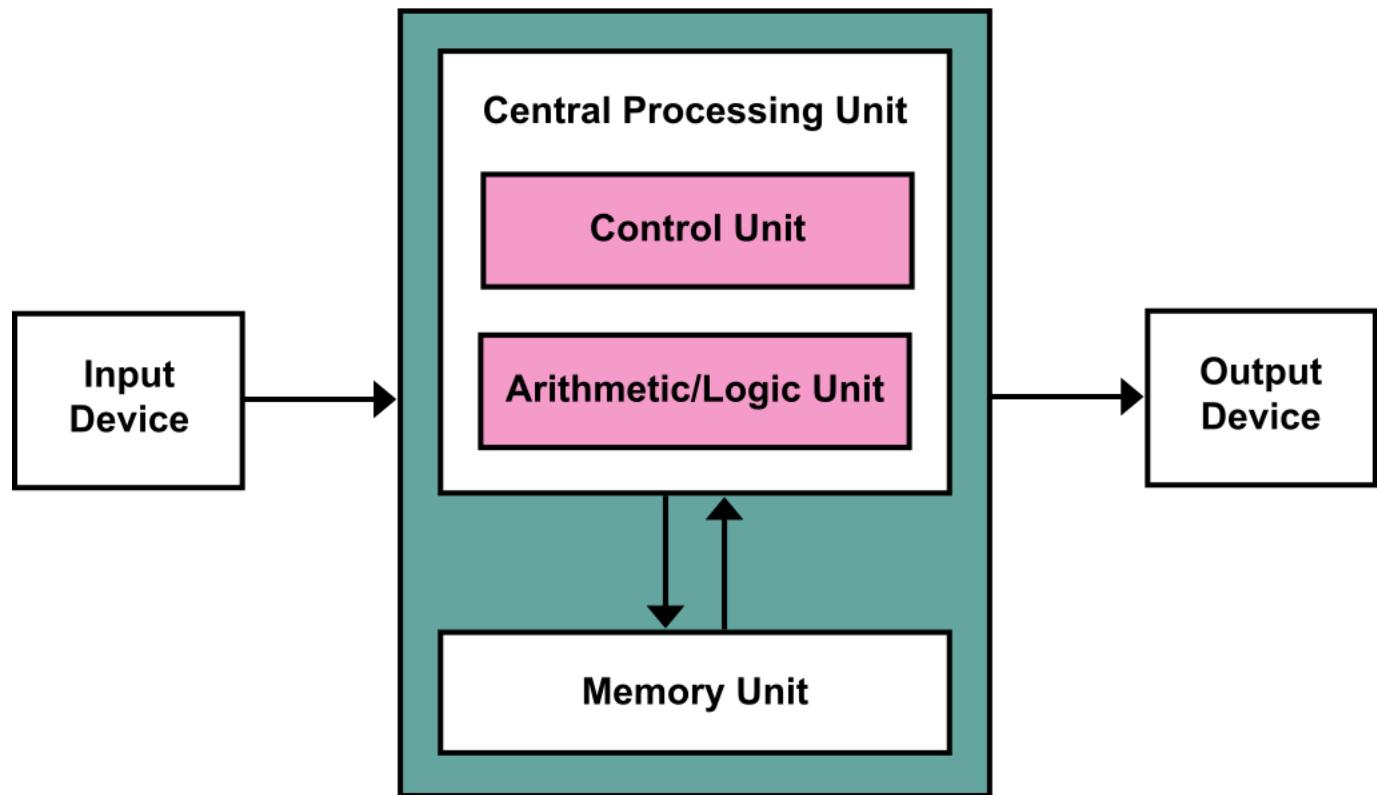
Mais... le nombre de portes croît
exponentiellement (la taille et l'énergie avec).

2^n

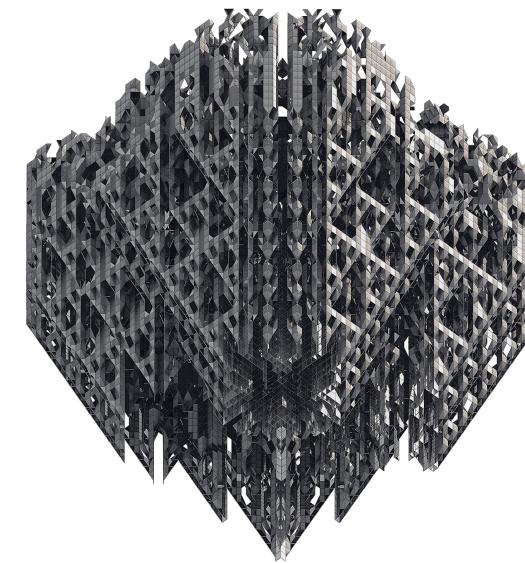
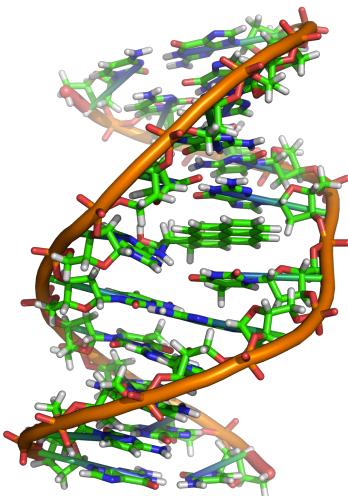
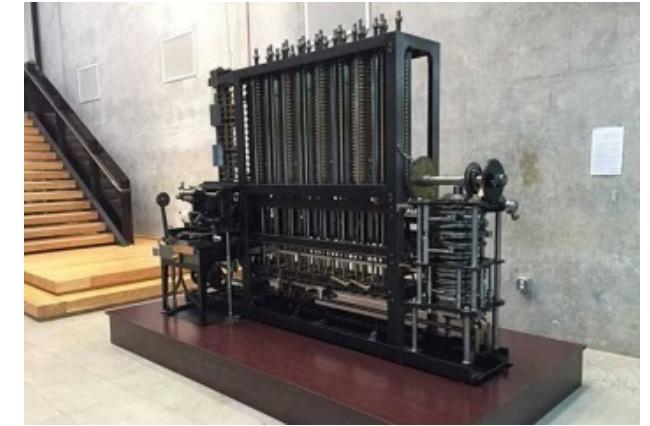
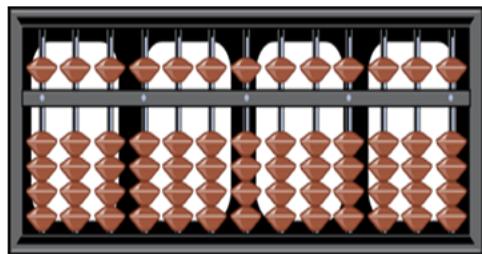


nos ordinateurs “classiques”

L'architecture de Von Neuman (Mémoire + ALU) permet de contourner le problème de la taille du circuit, mais repose le problème du temps de calcul



Autres modèles

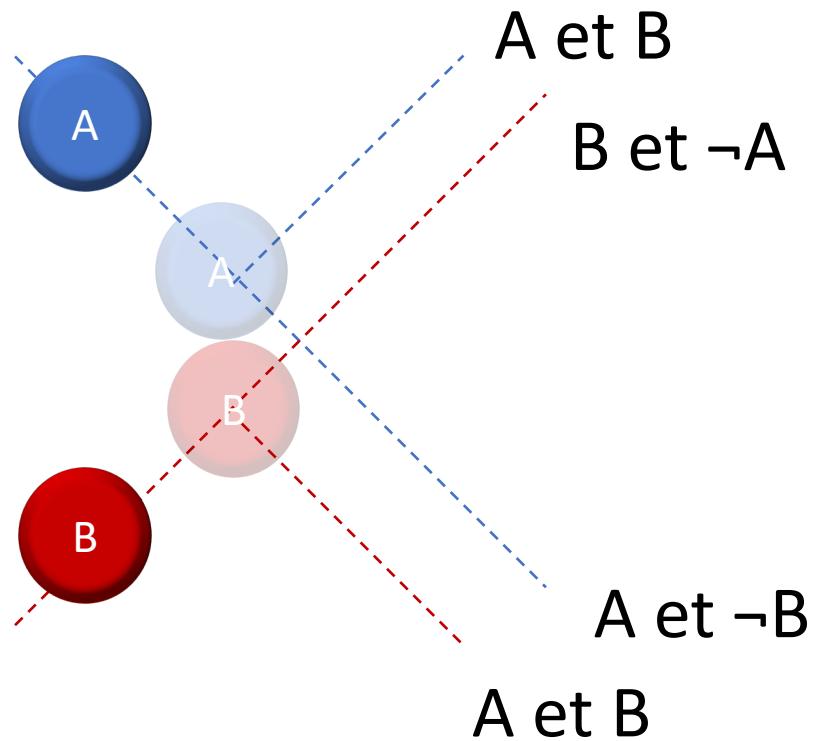
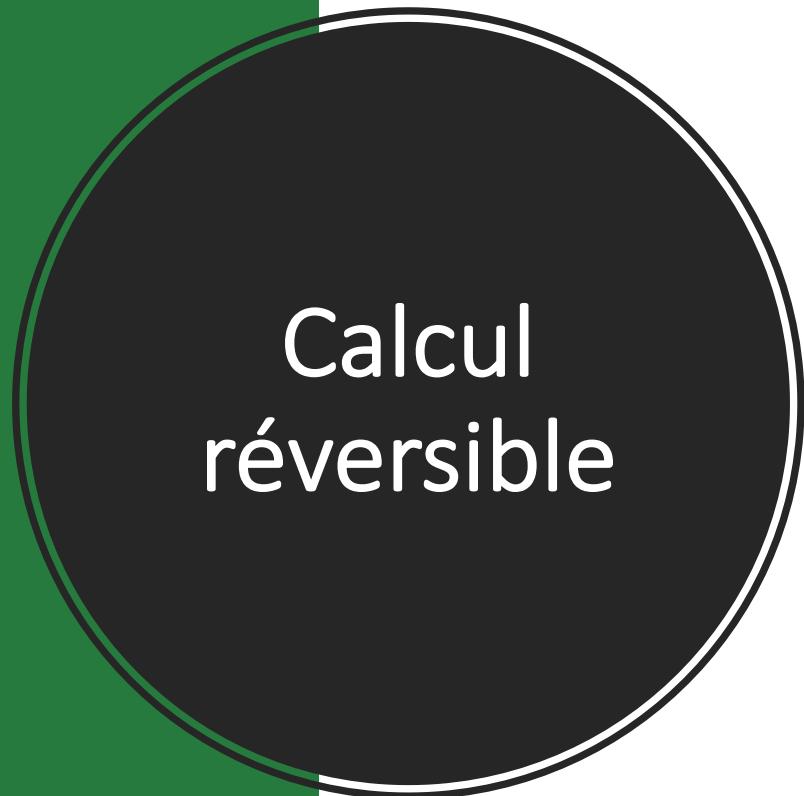


Q

calcul réversible

Ordinateur Mécanique Réversible : « boules de billard » (BBM : Billiard Ball Machine)

Edward Fredkin & Tommaso Toffoli - 1982



Calcul réversible

Ordinateur Mécanique Réversible : « boules de billard » (BBM : Billiard Ball Machine) Edward Fredkin & Tommaso Toffoli - 1982

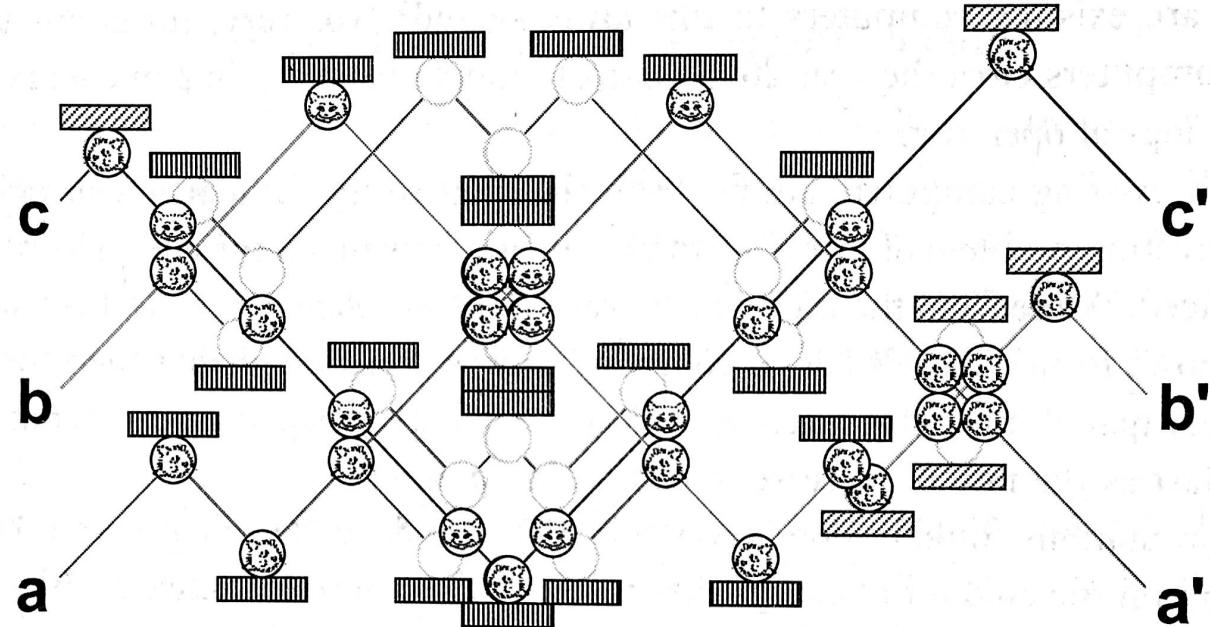


Figure 3.14. A simple billiard ball computer, with three input bits and three output bits, shown entering on the left and leaving on the right, respectively. The presence or absence of a billiard ball indicates a 1 or a 0, respectively. Empty circles illustrate potential paths due to collisions. This particular computer implements the Fredkin classical reversible logic gate, discussed in the text.

Michael Nielsen and Isaac Chuang (2000). *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press. ([ISBN 0-521-63503-9](#)).

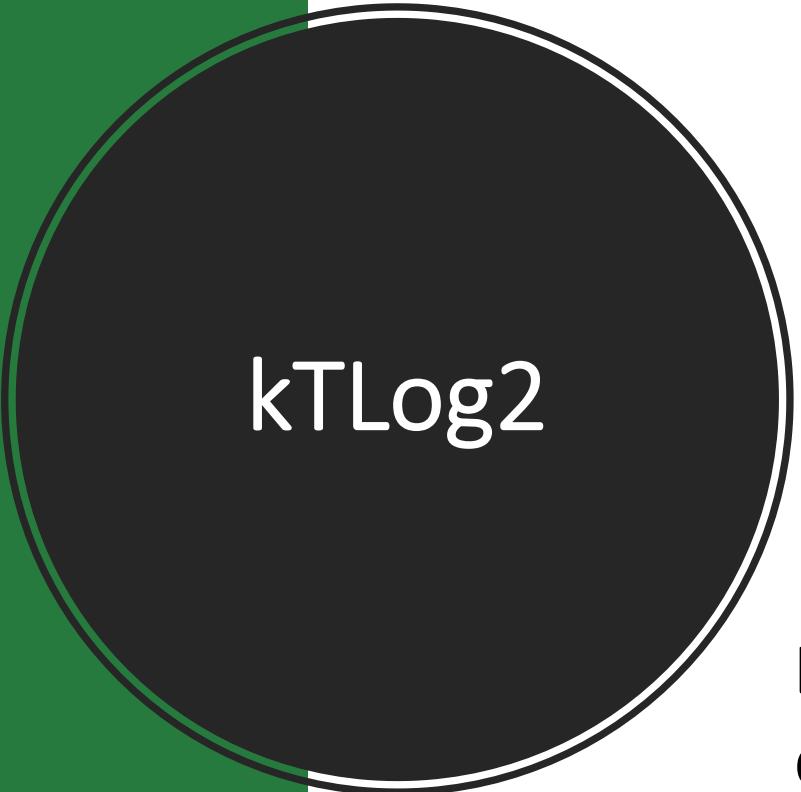
Calcul et énergie :

Principe de Landauer (première forme) : Lorsqu'un ordinateur efface un bit d'information, la quantité d'énergie dissipée dans son environnement est au minimum $kT\log 2$

Principe de Landauer (seconde forme) : Lorsqu'un ordinateur efface un bit d'information, l'entropie de son environnement augmente au minimum de $kT\log 2$:

$$3 \cdot 10^{-21} \text{ J}$$

Les ordinateurs actuels se situent à environ 500 fois cette limite.



$kT\log 2$

Le calcul réversible, c'est le cas du calcul quantique n'est pas soumis à cette limite.

complexité algorithmique

complexité concept et calcul

La complexité en informatique : $f(\text{espace}, \text{temps}, \text{énergie})$

Pour un problème P, si on prend le cas de la complexité en temps, qui varie comme le nombre d'instructions qu'il faut pour résoudre le problème en fonction d'un paramètre n qui le dimensionne

Soit $C(n)$ cette complexité ou ce nombre d'instructions, on va chercher à évaluer $C(n)$ lorsque $n \rightarrow \text{l'infini}$.

Une méthode consiste à regarder comment $c(n+1)$ varie par rapport à $c(n)$:

Variation de $C(n)$	Complexité	Notation « grand O »
$C(n+1) = C(n)$	constante	$\mathcal{O}(1)$
$C(n+1) = C(n) + \varepsilon$ (par exemple $C(n+n) = C(n) + 1$)	Logarithmique	$\mathcal{O}(\log(n))$
$C(n+1) = C(n) + k$	Linéaire (kn)	$\mathcal{O}(n)$
$C(n+1) = C(n) + n$	Polynomiale	$\mathcal{O}(n^2)$
$C(n+1) = C(n) + n^k$	Polynomiale	$\mathcal{O}(n^{k+1})$
$C(n+1) = C(n)*2$	Exponentielle	$\mathcal{O}(2^n)$
$C(n+1) = C(n)*n$	Exponentielle	$\mathcal{O}(n!)$
...		

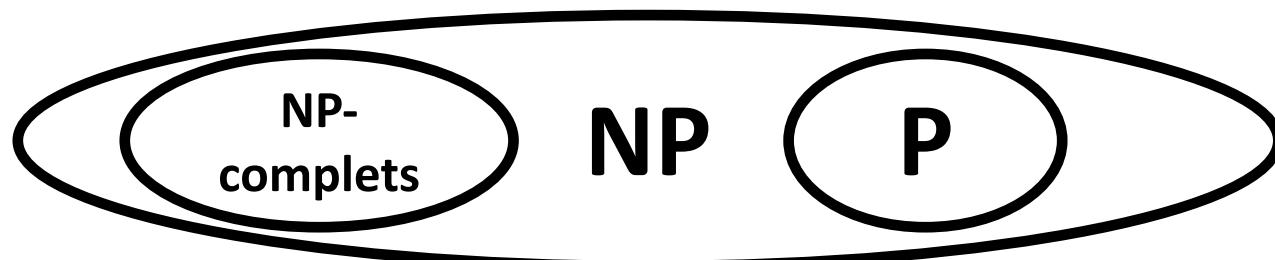
complexité exemples

Distribution de cartes à n joueurs:

Méthode de distribution	\mathcal{O}	exemple
Il y a 52 cartes à distribuer, on les distribue, \forall le nombre de joueurs.	$\mathcal{O}(1)$	Lire les 10 premières lignes d'une liste.
Distribuer 1 carte, diviser le nombre de joueurs par 2 et distribuer une autre carte, diviser à nouveau le nombre de joueurs par 2 et distribuer une autre carte, et ainsi de suite.	$\mathcal{O}(\log_2(n))$	Chercher un élément dans une liste ordonnée
Distribuer 2 cartes à chaque joueur	$\mathcal{O}(n)$	Chercher un élément dans une liste non ordonnée
Distribuer n cartes à chaque joueur, ou distribuer n cartes au $n^{\text{ième}}$ joueur.	$\mathcal{O}(n^2)$	Traitement de chaque paire d'une liste.
Distribuer une carte au premier joueur, 2 cartes au 2 nd , 4 cartes au 3 ^{ième} , 8 au 4 ^{ième} , et ainsi de suite : à chaque joueur deux fois plus de cartes qu'au joueur précédent.	$\mathcal{O}(2^n)$	Lorsqu'il faut traiter chaque sous-ensemble d'un ensemble, ou parcourir un arbre binaire

P et NP sont deux ensembles de problèmes de décisions

- Un problème p peut-être résolu de manière efficace si il existe un algorithme d'ordre polynomial au plus qui résout p. Alors p est dans P.
- Il existe des problèmes pour lesquels on connaît un algorithme exponentiel qui les traite, mais on ne sait pas s'il existe un algorithme polynomial qui les résout. Mais si il existe une manière de vérifier leur solution en un temps au plus polynomial alors ces problèmes sont de classe NP.
- On ne sait pas si $P = NP$.
- On sait qu'il existe une catégorie de problèmes NP qui concentrent en eux toute la difficulté des problèmes NP. On les appelle NP-complets.
- Si l'on considère un seul de ces problèmes NP-complets :
 - soit on arrive à prouver qu'il n'existe pas d'algorithme polynomial le résolvant, alors $P \neq NP$
 - Soit on trouve un algorithme polynomial qui le résout, alors $P = NP$.



P vs NP

Exemple 3SAT

Le problème « 3SAT » est un problème dit de satisfiabilité:

C'est-à-dire : pour une expression logique F de n booléens B_i , existe-t-il une combinaison des valeurs de B_i , telle que $F = \text{Vraie}$?

En particulier le 3SAT porte sur les expressions de la forme:

(par exemple)

$$F = (B_0 + B_3 + B_4) * (\neg B_1 + B_5 + \neg B_7) * (B_2 + B_3 + \neg B_4) * \dots * (\neg B_0 + B_1 + B_6)$$

La résolution passe par le calcul de F pour toutes les combinaisons de valeurs des n booléens.

Il s'agit d'un problème NP-complet

A noter que le problème 2SAT est de classe P
 $(F = (B_0 + B_3) * (\neg B_1 + B_2) * (B_2 + B_3) * \dots * (\neg B_0 + B_4))$

Exemples de NP complets

CLIQUE (théorie des graphes): une clique dans un graphe non directionnel est un ensemble de sommets tous reliés entre eux par une arête. Sa taille est le nombre de ses sommets. Pour un graphe G, existe-t-il une clique de taille k ?

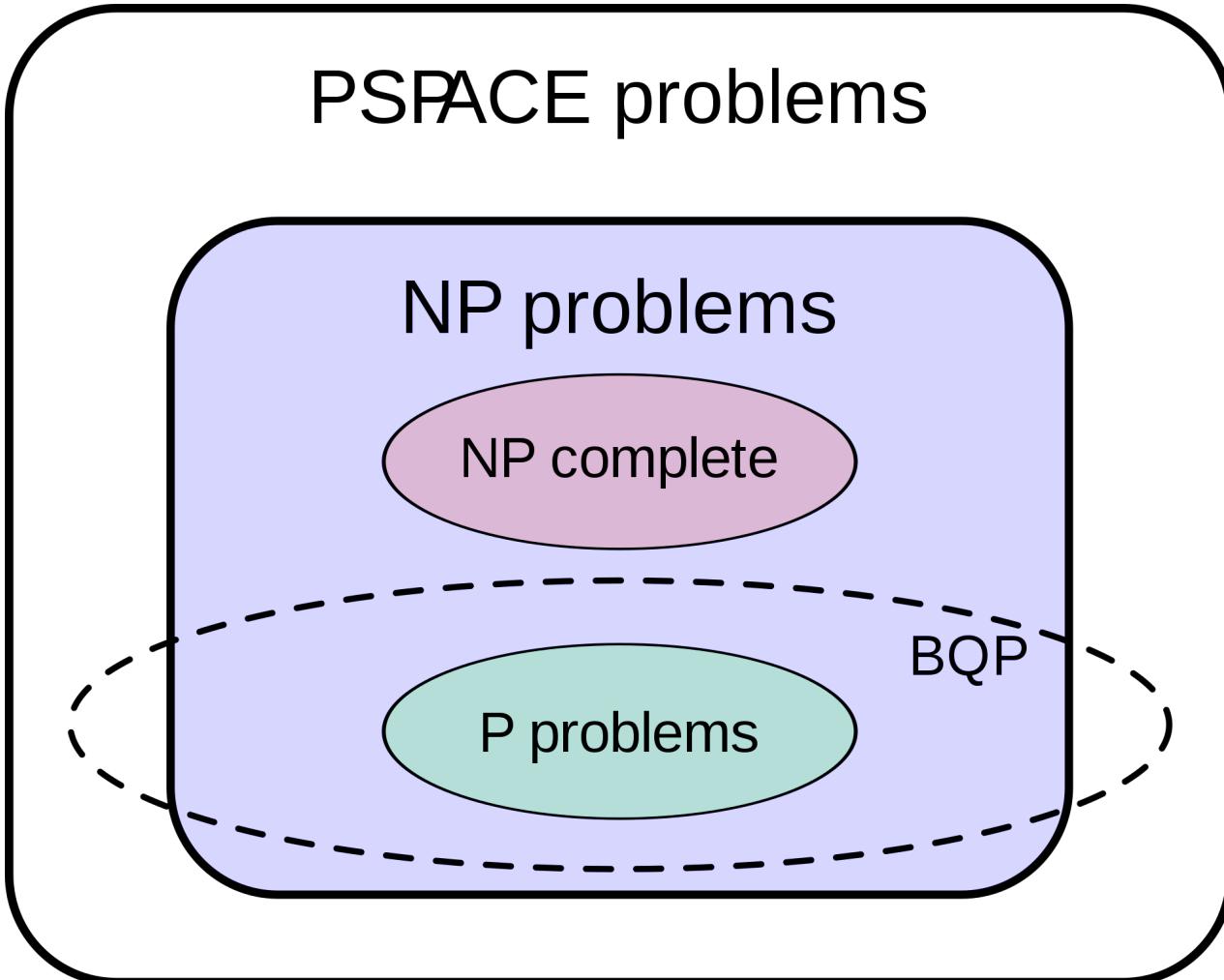
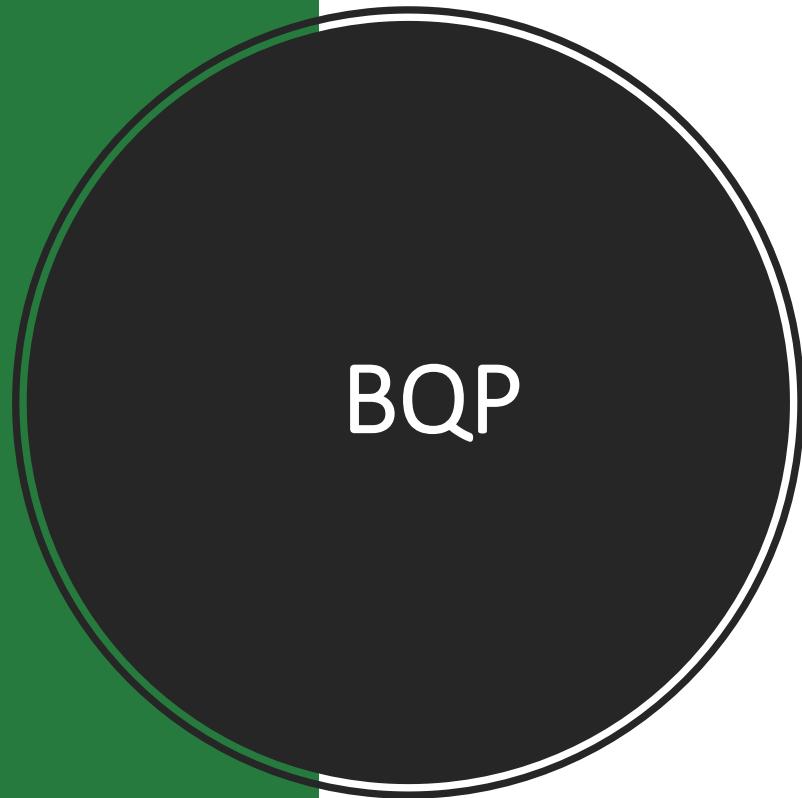
SOMME de SOUS ENSEMBLE (arithmétique) : Etant donné E un ensemble fini d'entiers, et un entier s, existe-t-il un sous ensemble de E dont la somme des éléments vaut s ?

COUPE MAXIMALE (MAXCUT, théorie des graphes) Étant donné un graphe, et des poids sur les arêtes, une coupe peut-être décrite comme un ensemble de sommets, et le poids de la coupe est alors la somme des poids des arêtes ayant une extrémité à l'intérieur de cet ensemble et l'autre à l'extérieur. Une coupe est maximum si son poids est maximum (parmi toutes les coupes).

SAC A DOS (Knapsack Problem, optimisation combinatoire) : Etant donnés des objets dont on connaît le poids et la valeur : il s'agit de trouver la combinaison d'objets à mettre dans le sac qui maximise la valeur totale sans dépasser un poids maximum.

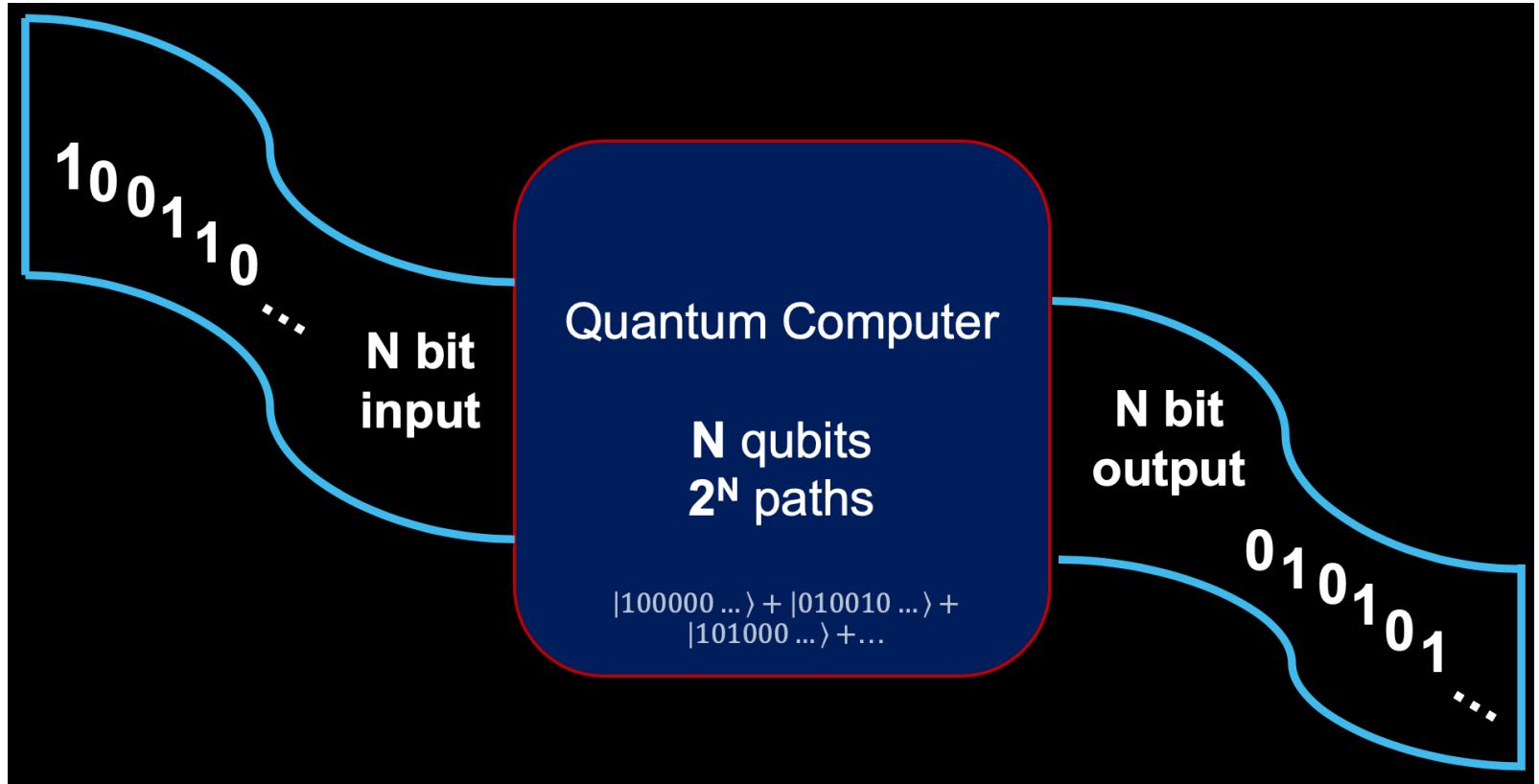
https://fr.wikipedia.org/wiki/Liste_de_probl%C3%A8mes_NP-complets

« Les relations supposées entre BQP et les autres classes de complexité. »



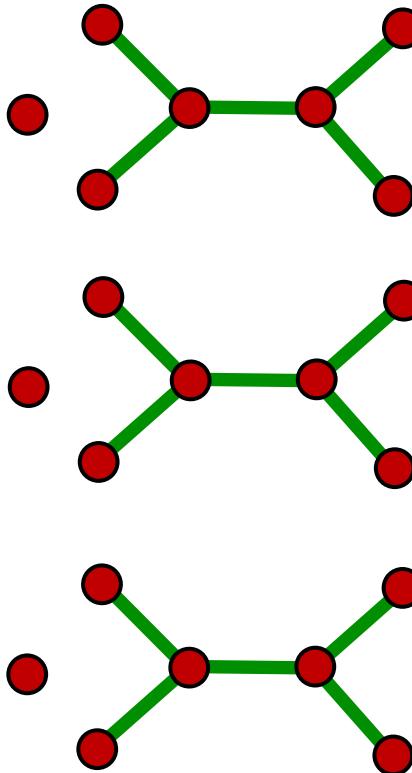
Michael Nielsen and Isaac Chuang (2000). *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press. ([ISBN 0-521-63503-9](#)).

BQP



Retour sur les Graphes Théorème des mineurs

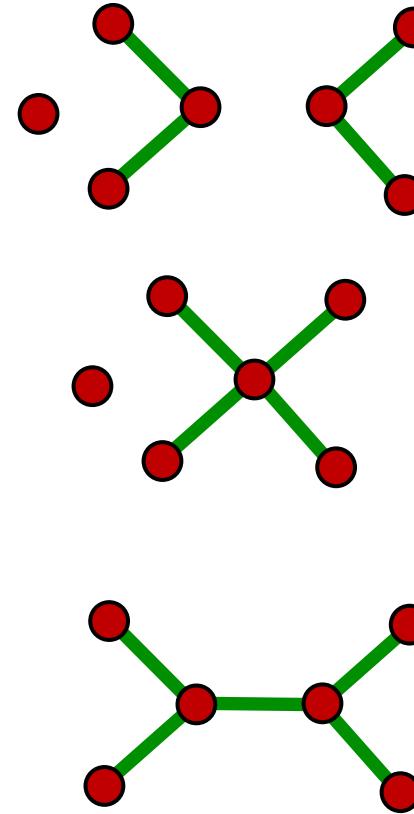
Mineur : le graphe G' est un *mineur* du graphe G , si en partant de G et en le réduisant par les opérations (a), (b) et (c) on obtient G' .



(a) Suppression
d'une arête

(b) Contraction
d'une arête (les
nœuds au sommet
de l'arête choisie
sont fusionnés)

(c) Suppression
d'un nœud isolé



Théorème des mineurs : Dans tout ensemble infini de graphes,
il en existe un qui est le mineur d'un autre.
(Robertson et Seymour)

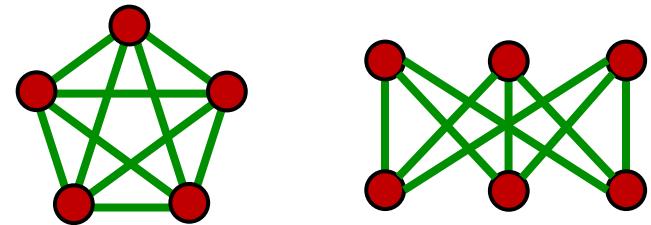
Retour sur les Graphes Théorème des mineurs

Un ensemble E de graphes est dit stable par minoration si tout mineur d'un graphe de E est lui-même dans E.

Seconde forme du théorème de Roberston et Seymour:

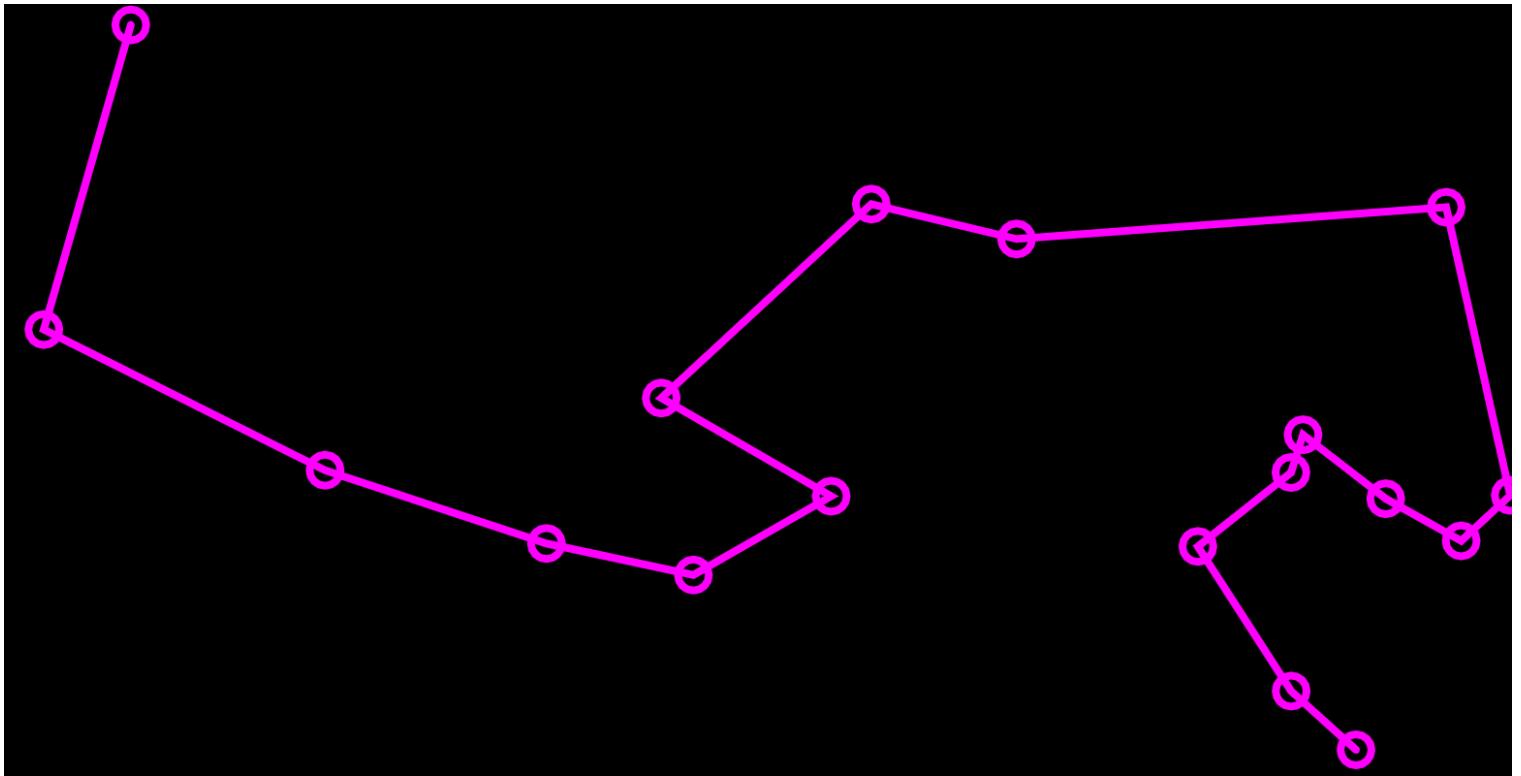
Tout ensemble E de graphes stable par minoration est définissable comme l'ensemble des graphes n'ayant pour mineur aucun des graphes d'une liste finie, caractéristique de E, et appelée « ensemble d'obstruction de E »

par exemple, l'ensemble des graphes planaires sont les graphes n'ayant aucun des graphes K_5 et $K_{3,3}$ comme mineurs.



- ❖ Il a été démontré qu'il n'existe pas d'algorithme permettant de calculer l'ensemble des mineurs d'une classe définie de graphes
- ❖ Par exemple sur le tore on ne sait pas déterminer l'ensemble d'obstruction des graphes qu'on peut y dessiner sans croisement d'arêtes (on en a trouvé 16629, mais sans doute parmi d'autres).
- ❖ On arrive à prouver qu'il existe un algorithme en $\mathcal{O}(n^3)$ qui indique quels sont les graphes que l'on peut dessiner sur un tore sans croisement d'arêtes, mais l'algorithme n'a pas été découvert.

Démo
“Voyageur de
Commerce”
(TSP)



Youtube : The Coding Train. Genetic Algorithms