Informatique théorique : Construire des arbres dans le monde de Tetris et de Minecraft

Alexandre Blondin Massé

Département d'informatique
Université du Québec à Montréal

16 novembre 2016
Midi-conférence du département d'informatique
Université de Montréal

Table des matières

Informatique théorique : qu'est-ce que c'est ?

- Discipline à l'intersection des **mathématiques** et de l'**informatique**;
- S'intéresse à la **classification** des problèmes selon leur **difficulté** :
  - Problèmes **polynomiaux** : peuvent être résolus en temps $\mathcal{O}(n^k)$;
  - Problèmes **exponentiels** : peuvent être résolus en temps $\Omega(k^n)$.
  - Problèmes **indécidables** : il est impossible de construire un algorithme pour résoudre le problème.

Quelques problèmes célèbres

- Problème de la **fouille**. $\mathcal{O}(n)$
- Problème du **tri**; $\mathcal{O}(n \log n)$
- Recherche d'un **mot** dans un **texte** ? $\mathcal{O}(n + m)$
- Multiplication de **matrices**; $\mathcal{O}(n^{2.3728639})$
- Test de **primalité**. Le plus **grand nombre premier** connu est

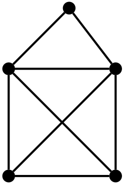
  $$2^{74\ 207\ 281} - 1 \quad \text{(22 338 618 chiffres)}$$

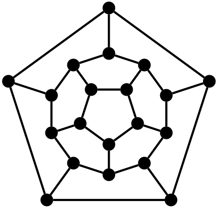
  $\widetilde{\mathcal{O}}(\log^6 n)$
- Décomposition en **facteurs premiers**. À la base d'algorithmes de **cryptographie**. Ω(polynôme) mais o(exponentiel).

Quelques problèmes célèbres (2/2)

Prenons un graphe de n **sommets** et m **arêtes** :

- Problème du **plus court chemin** dans un graphe :
  - Sans **poids** sur les arêtes; $\mathcal{O}(n + m)$
  - Avec **poids** sur les arêtes. $\mathcal{O}(m \log n)$
- Problème du **graphe eulérien/hamiltonien**; $\mathcal{O}$(non polynomial)





La classe NP

- **Classe de problèmes** très importante en théorie de la **complexité**;
- NP $\neq$ *nonpolynomial*;
- NP $=$ *nondeterministic polynomial*;
- Intuitivement, il s'agit des problèmes qu'on peut espérer **résoudre** plus ou moins efficacement.
- Plus formellement, un problème est **de classe NP** si, pour tout exemplaire, on peut lui associer
  - un **certificat** de taille polynomiale et
  - un **algorithme de vérification** polynomial qui valide ce certificat.

Exemples

- Est-ce que n est un nombre **composé** ?
  - **Certificat** : deux nombres a et b;
  - **Vérification** : est-ce que $ab = n$?
- Est-ce qu'une chaîne s apparaît dans un texte t ?
  - **Certificat** : deux indices i et j;
  - **Vérification** : est-ce que $s = t[i, j]$?
- Est-ce que G est un graphe **hamiltonien** ?
  - **Certificat** : une suite de **sommets** de G;
  - **Vérification** : est-ce la suite forme un **cycle** passant par chaque sommet exactement une fois.

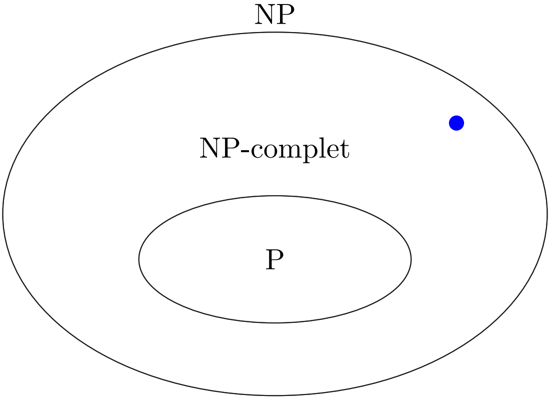
Problèmes NP-complet

Exemples de problèmes **NP-complets** :

- Étant donné une formule logique, est-elle **satisfiable** ?
- Étant donné un graphe, possède-t-il un **cycle hamiltonien** ?
- Étant donné un graphe reliant n villes, existe-t-il une **tournée** de poids au plus k ?

À ce jour, on ne sait pas s'il est possible de résoudre ces problèmes en temps **polynomial**.

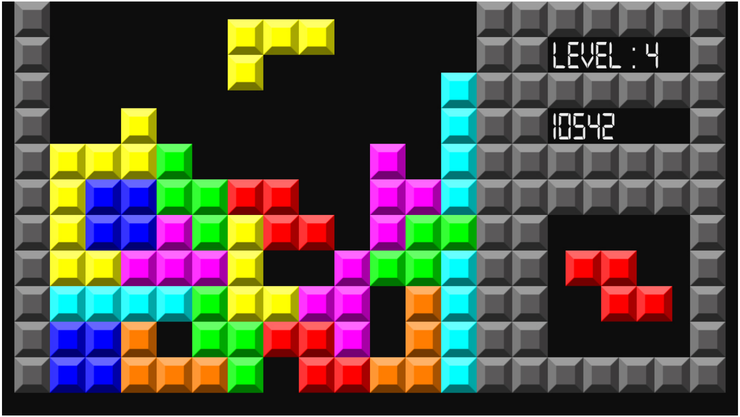
Est-ce que P = NP ?



- Un des problèmes du **millénaire** !

Table des matières

Tetris

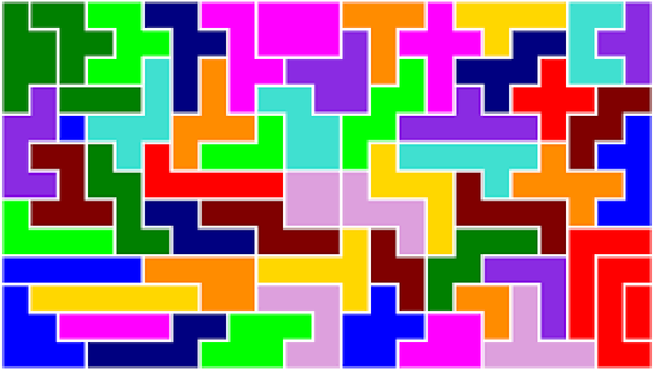


(source: The Tetris Company)

Dominos

(source: apkpure.com)

Polyominos

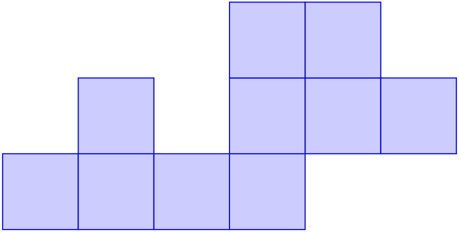


(source: David J. Goodger)

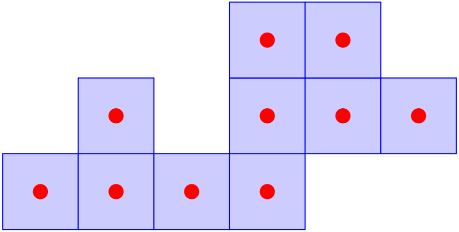
Polyominos (suite)

- Ils ont été beaucoup étudié par les **chercheurs**;
- Leur structure est très **complexe**;
- Beaucoup de problèmes **difficiles** :
  - Combien y a-t-il de **polyominos** de n cellules ? **Inconnu pour** $n \geq 57$;
  - Peut-on **paver** le plan à l'aide d'un ensemble de **polyominos** donné ? **Indécidable en général**.
  - Peut-on **paver** une **région finie** donnée avec un ensemble de polyominos ? **NP-complet**.

Dans chaque polyomino se cache un graphe

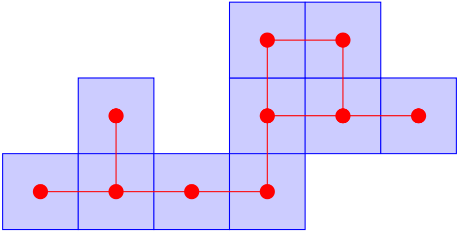


Dans chaque polyomino se cache un graphe



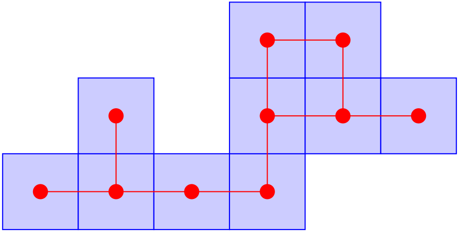
- Les **sommets** sont les **cellules**;

Dans chaque polyomino se cache un graphe



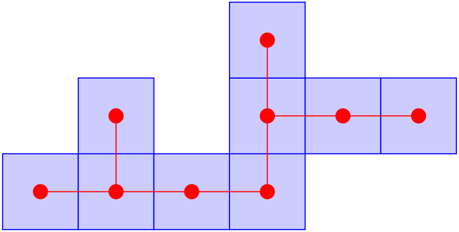
- Les **sommets** sont les **cellules**;
- Une **arête** relie deux cellules si elles partagent un côté.

Dans chaque polyomino se cache un graphe



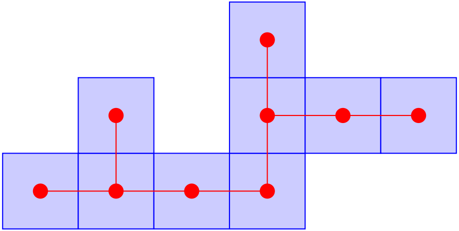
- Les **sommets** sont les **cellules**;
- Une **arête** relie deux cellules si elles partagent un côté.
- Un tel graphe peut contenir des **cycles**.

Polyominos arbres



- S'il n'y a pas de **cycle**, on parle de **polyomino-arbre**.
- Ce polyomino-arbre possède 4 **feuilles**.

Nombre maximal de feuilles ?



- Considérons un polyomino-arbre de n **cellules**;
- Quel est le nombre **maximal** de feuilles qu'il peut posséder ?

Polyominos-arbres pleinement feuillus

Théorème [B. M., Goupil, De Carufel, Samson; 2016]

Pour tout entier $n \geq 1$, le nombre maximal de feuilles que peut posséder un **polyomino**-arbre est donné par

$$\ell_2(n) = \begin{cases} 0, & \text{si } n = 1; \\ 2, & \text{si } n = 2; \\ n - 1, & \text{si } n = 3, 4, 5; \\ \ell_2(n-4) + 2, & \text{si } n \geq 6. \end{cases}$$

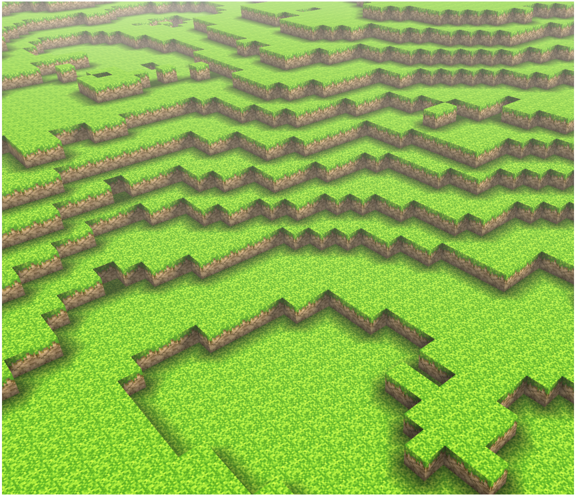
Démonstration

Par **construction** et par **contre-exemple minimal**.

Table des matières

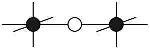
Minecraft

- Voir exemple en WebGL;



Polycubes

- La version 3D des polyominos est appelée **polycube**;
- Chaque cellule a au plus 6 **voisins** plutôt que 4;
- La question se pose aussi : quel est le nombre **maximal** de feuilles qu'on peut réaliser dans un **polycube-arbre** ?



Polycubes-arbres pleinement feuillus

Théorème [B. M., Goupil, De Carufel, Samson; 2016]

Pour tout entier $n \geq 1$, le nombre maximal de feuilles que peut posséder un **polycube**-arbre est donné par

$$\ell_3(n) = \begin{cases} 0, & \text{si } n = 1; \\ f_3(n) + 1, & \text{si } n = 6, 7, 13, 19, 25; \\ f_3(n), & \text{si } 0 \leq n \leq 40 \text{ et } n \neq 1, 6, 7, 13, 19, 25; \\ f_3(n - 41) + 28, & \text{si } 41 \leq n \leq 81; \\ \ell_3(n - 41) + 28, & \text{si } n \geq 82. \end{cases}$$

où

$$f_3(n) = \begin{cases} \lfloor (2n+2)/3 \rfloor, & \text{si } 0 \leq n \leq 11; \\ \lfloor (2n+3)/3 \rfloor, & \text{si } 12 \leq n \leq 26; \\ \lfloor (2n+4)/3 \rfloor, & \text{si } 27 \leq n \leq 40. \end{cases}$$

Quelques polycubes pleinement feuillus

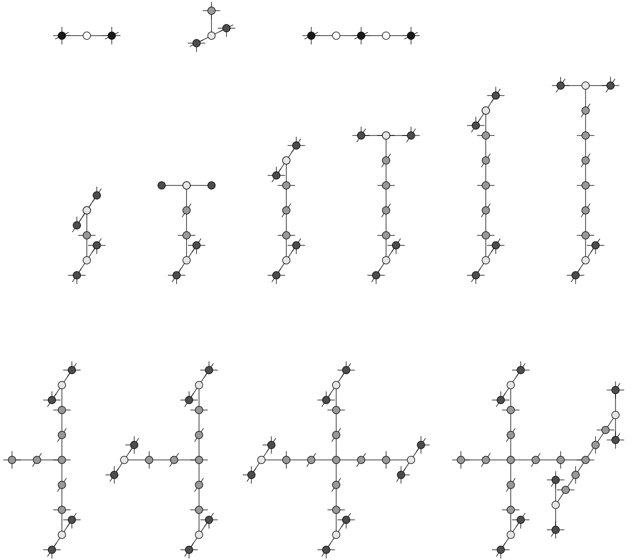


Table des matières

À quoi ça sert ?

- Pour être honnête, probablement **à rien**;
- Mais...
  - ... c'est **amusant** (selon le point de vue);
  - ... il semblerait que ça puisse être intéressant en **chimie**.
- Plus précisément, on peut se demander si ces **structures** apparaissent naturellement au niveau **moléculaire** ou dans les réseaux **cristallins**.

La suite ?

- Étendre les résultats à d'**autres réseaux**;
- Étudier le problème sur des **graphes quelconques**, pas nécessairement **géométriques**.

Conjecture

Le problème de décider si un graphe simple quelconque possède un sous-arbre induit de n **cellules** et ℓ **feuilles** est **NP-complet**.