La complexité de certains JEUX

Cryptarithmétique

Un grand classique :

```
  SEND
 +MORE
 -----
 MONEY
```

Cryptarithmétique

Un grand classique :

```
 SEND
+MORE
-----
MONEY
```

M ne peut être autre que 1 !

Cryptarithmétique

Un grand classique :

```
  SEND
+1ORE
-----
 1ONEY
```

S doit être 8 ou 9, donc O à son tour au plus 1, et comme 1 occupé donc 0 !

Cryptarithmétique

Un grand classique :

```
 SEND
+1ORE
-----
1ONEY
```

N doit être E+1 (retenue) et donc pas de retenue de la colonne

Ainsi S doit être 9

Cryptarithmétique

Un grand classique :

```
  9END
+10RE
-----
 10NEY
```

N=E+1, mais comme N+R+ε=E (ε - retenue)

R+ε=9 et comme 9 occupé, donc R=8 et ε=1

Cryptarithmétique

Un grand classique :

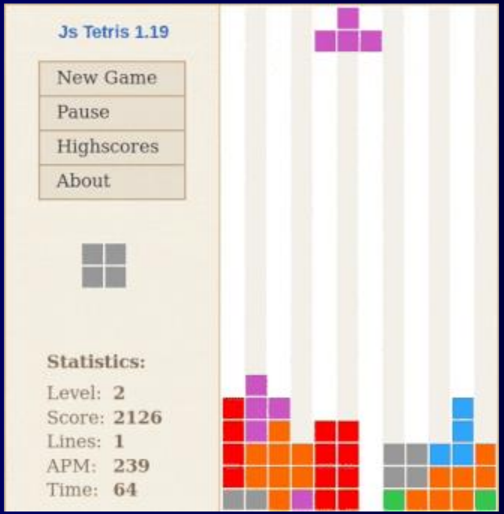
```
  9END
+108E
-----
10NEY
```

N = E+1, il nous restent donc pour (N,E)

(7,6), (6,5), (5,4), (4,3), (3,2)

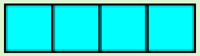
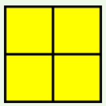
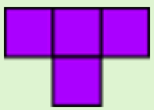
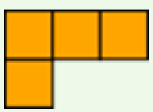
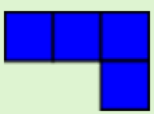
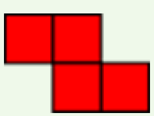
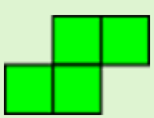
A vous de continuer (après le cours… :=))

Tetris est un jeu vidéo de puzzle conçu par Alekseï Pajitnov en 1985.



Préliminaires

- Le jeu se déroule sur un rectangle.
- Des pièces tombent du haut vers le bas (blocks – il sont détaillés dans la page suivante).
- On a un block (aléatoire) qui tombe à la fois.
- On peut tourner les blocks et les déplacer à gauche ou à droite.
- Les lignes remplies disparaissent.
- Si une colonne est rempli jusqu'en haut le jeu se termine (*la mort*).

Forme	Noté	Construction
	I	Quatre carrés alignés.
	Sq	Carré de 2x2.
	T	Trois carrés en ligne et un carré sous le centre.
	RG	Trois carrés en ligne et un carré sous le côté gauche.
	LG	Trois carrés en ligne et un carré sous le côté droit.
	LS	Carré de 2x2, dont la rangée supérieure est glissée d'un pas vers la gauche.
	RS	Carré de 2x2, dont la rangée supérieure est glissée d'un pas vers la droite.

Préliminaires

Pour pouvoir parler de complexité du problème on est amenés à passer à une version *information parfaite* :

- on connaît à l'avance la suite entière des blocks
- on a une position initiale donnée

La NP-complétude

Théorème : Il est NP-complet de savoir si on peut rester vivant après *n* pièces.

Un problème NP-complet

(Garey & Johnson, 1975)

Nom : triplets-Partition

Données : Un nombre naturel T, un ensemble de nombres naturels $A=\{a_1,a_2, \ldots, a_{3s}\}$, tels que $T/4 < a_i < T/2$ pour tout $1 \leq i \leq 3s$, et tels que la somme des a_i est sT.

Question : Peut-on partitionner l'ensemble en s ensembles $A_1, \ldots, A_s$, tels que la somme des éléments de chaque ensemble est T ?

Remarques : 1. La condition sur la taille des a_i, nous assure que si on a une solution alors chaque A_i contient exactement 3 éléments.

2. Les données peuvent être en unaire et le problème reste NP-complet.

Deux types de NP-complétude

NP-complétude faible

(Weak NP-completeness)

Les problèmes qu'on peut résoudre en temps polynomial lorsque leurs entrées sont codées en unaire sont appelés problèmes *faiblement NP-complets*.

Le problème du sac à dos et ses variantes appartiennent à cette catégorie. En effet, ces problèmes peuvent être résolus par programmation dynamique en temps *pseudo-polynomial*.

Deux types de NP-complétude (2)

NP-complétude forte

(Strong NP-completeness)

Les problèmes qui restent NP-complets dans ce cas sont dits *fortement NP-complets*. C'est le cas de la plupart des problèmes classiques, comme SAT, CLIQUE, la coloration de graphe, etc.

Remarque : Les problèmes NP-complets non-numériques sont fortement NP-complets

Deux types de NP-complétude pour des problèmes numériques

Faiblement NP-complets (NP-complets)

Les données peuvent être exponentielles (2^{n^c}) dans la taille *n* des données et leurs encodage reste polynomial en *n* - $\log(2^{n^c})$.

Fortement NP-complets

NP-complets même si on se limite à des données dont la valeur est polynomiale en n – c.à.d. les nombres peuvent être encodés en unaire.

Notions algorithmiques correspondantes

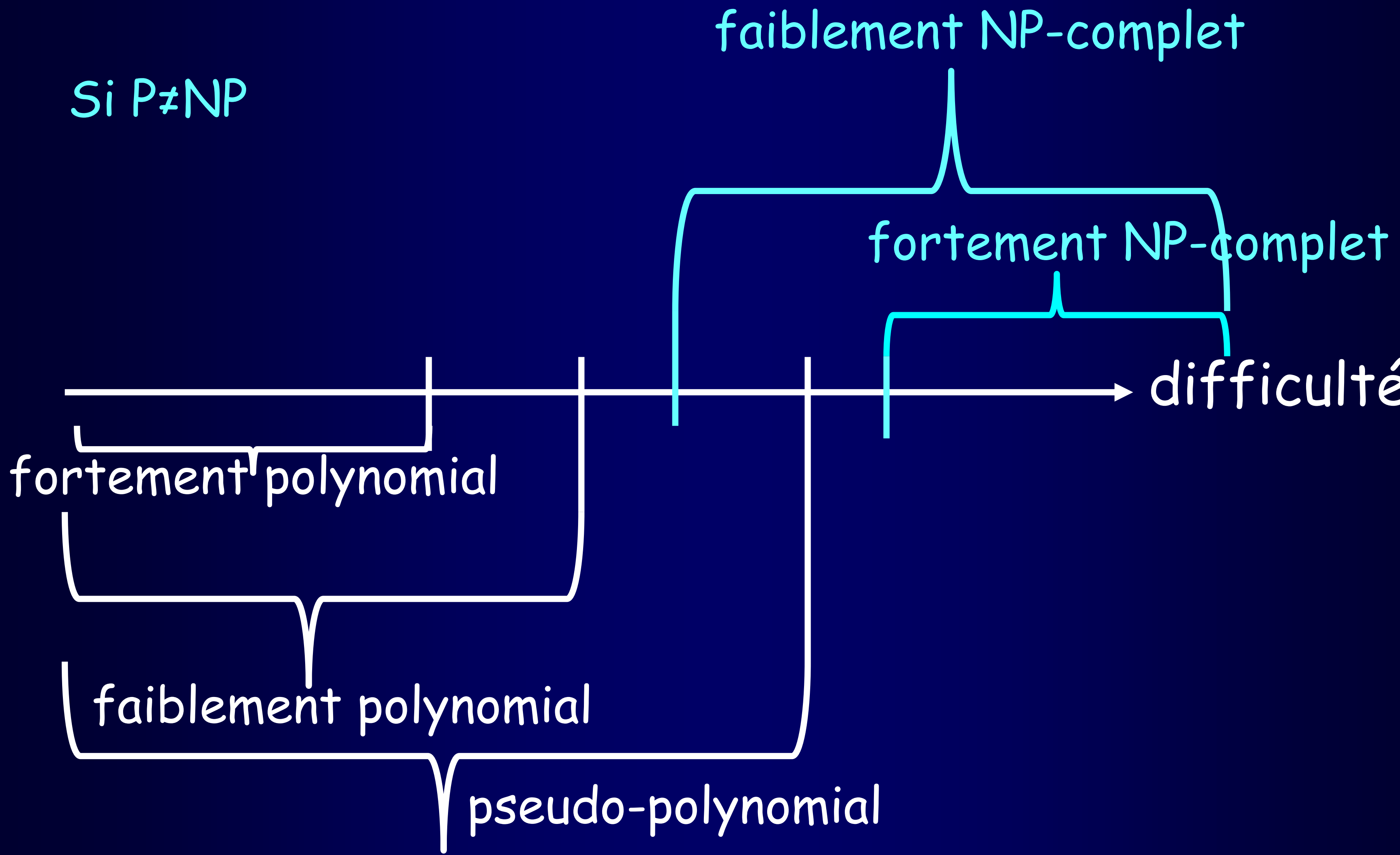
Pseudo-polynomial – polynomial en *n* et la plus grande donnée

Faiblement polynomial – polynomial en *n* et le log du plus grand donnée

Fortement polynomial – polynomial en *n*

Faiblement NP-complet exclut algorithme polynomial (si P≠NP) mais laisse possible pseudo-polynomial.

Tableau des difficultés



Revenons sur Tetris

On peut considérer différents objectifs dans le jeu

- maximiser le nombres de lignes supprimées
- maximiser le nombre de blocs qu'on a réussi de placer
- … etc

Lemme : pour toute fonction objectif vérifiable Tetris $\in$ NP.

On peut le prouver en choisissant de manière non-déterministe le placement de chaque pièce. Et la vérification se fait en temps polynomial …

La transformation

On a une donnée de triplets-Partition : $\{a_1, \ldots, a_{3s}, T\}$. On supposera que T est un multiple de 4. Les données sont sous une forme unaire (on peut le faire car la NP-complexité de triplets-Partition est forte).

Dans la transformation nous utiliserons uniquement les pièces

LG, RG, I, Sq.

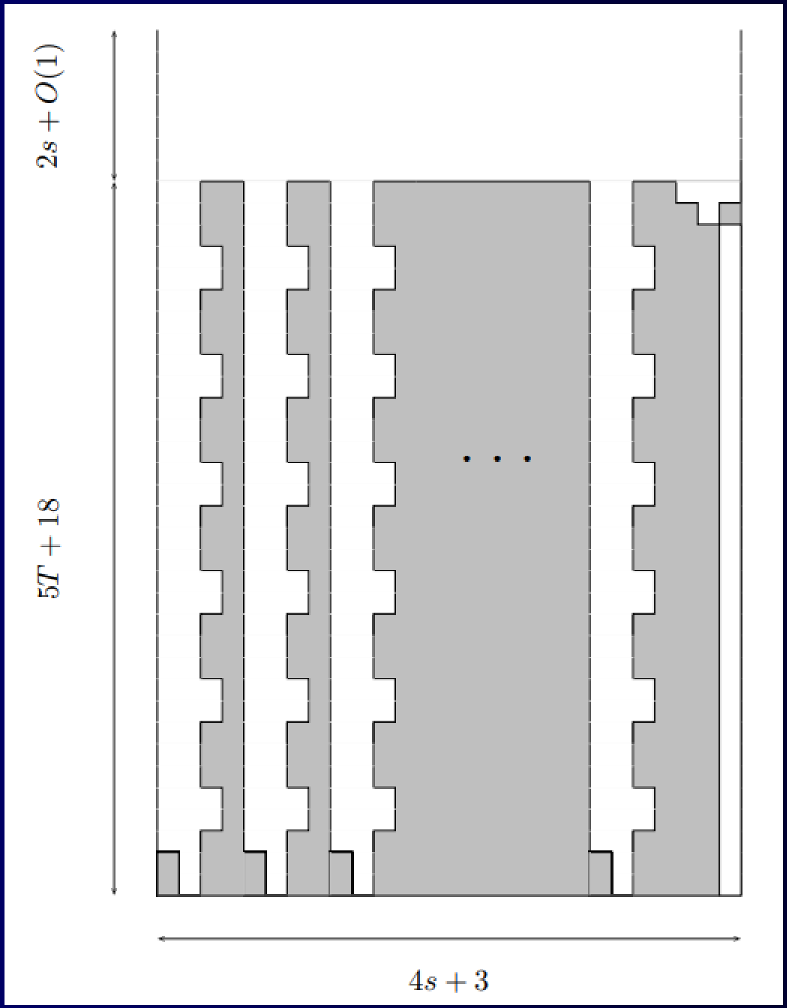
D'autres preuves existent aussi, en utilisant d'autres pièces.

Le tableau initiale

Largeur 4s+3, Hauteur 5T+18
(et il reste encore 2s+O(1) lignes en haut pour le cas où ...)

Intuitivement il y a la place pour s paquets qui correspondront aux ensembles A_i – ce seront les colonnes vides en blanc :
1-4, 5-8, ..., 4s-3-4s.

Les trois dernières colonnes joueront un rôle spécial.

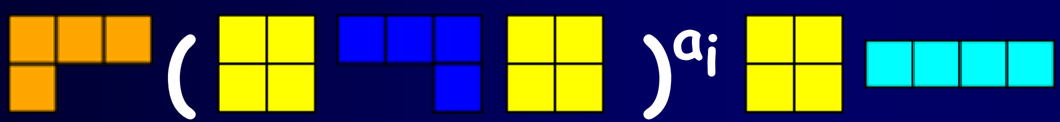


Les pièces

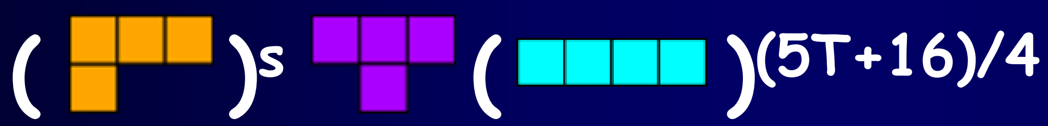
Etant donné a_i (codé en unaire)

$$f(a_i) = RG\ (Sq\ LG\ Sq)^{a_i}\ Sq\ I$$

c'est-à-dire



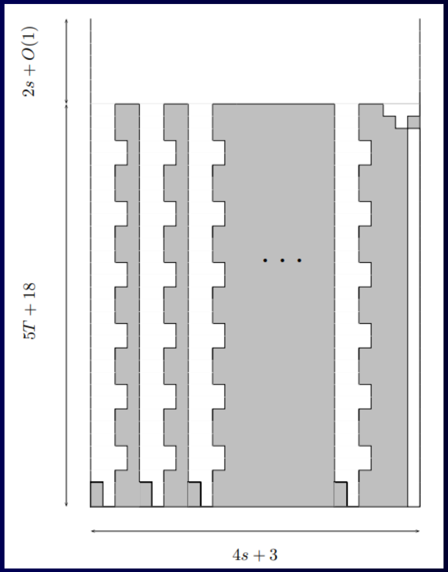
et à la fin $RG^s\ T\ I^{(5T+16)/4}$ c.à.d.



et toutes ces pièces arrivent dans cet ordre.

Quelques remarques

On peut remarquer que si on a une solution pour le problème triplets-Partition alors comme la somme de chaque A_i est la même, la hauteur des pièces correspondants à chaque A_i sera la même.



Les trois dernières colonnes constitueront une sorte de verrou, pour que les lignes ne disparaissent pas trop tôt.

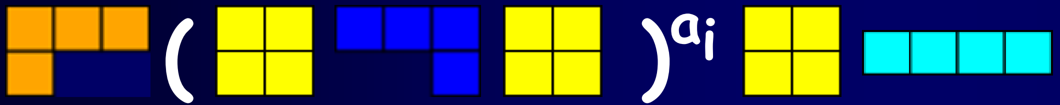
Transformation polynomiale

Le nombre total de pièces est

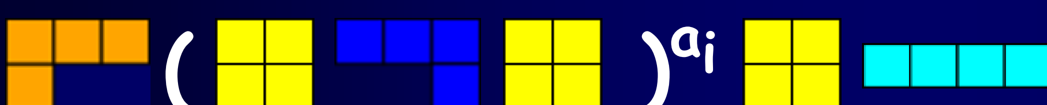
$$\sum_{i=1}^{3s}[1+3a_i+2]+s+1+\left(\frac{5T+16}{4}\right) \quad = \quad 10s+3sT+\frac{5T}{4}+5.$$

et ainsi comme triplets-Partition est donné en unaire, la construction est polynomiale.

Les pièces qui correspondent à un a_i

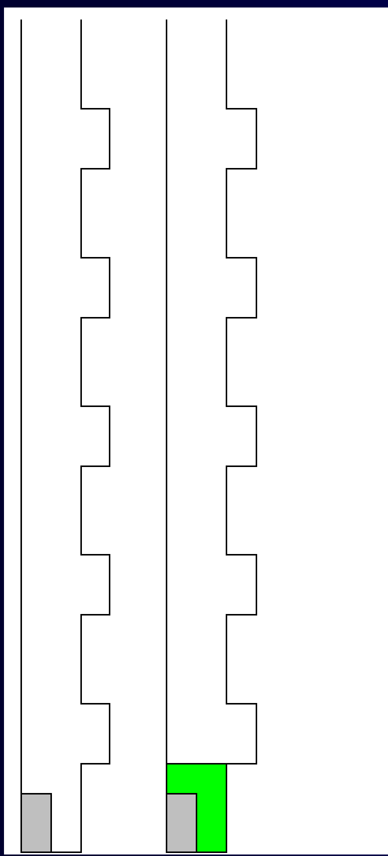


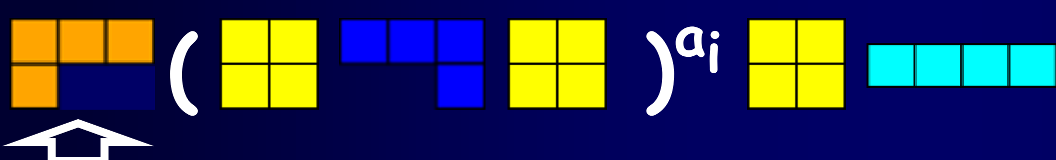
Les pièces qui correspondent à un a_i



0

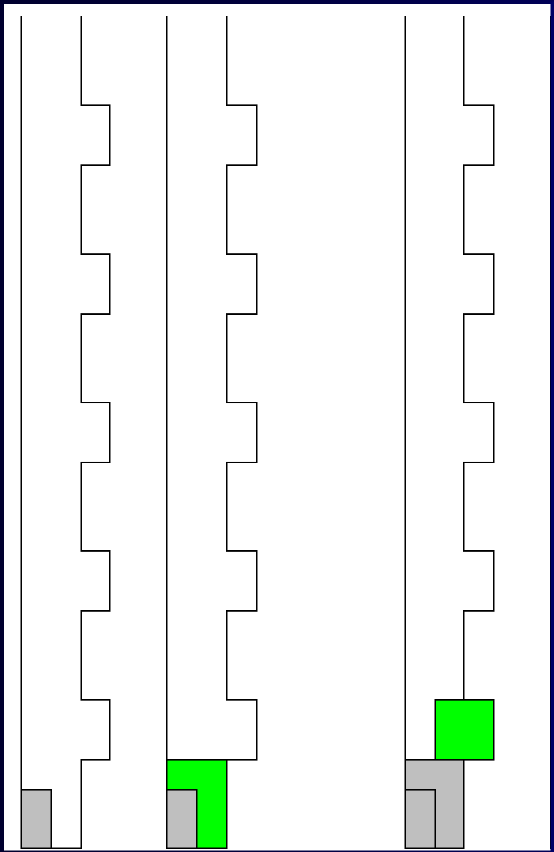
Les pièces qui correspondent à un a_i

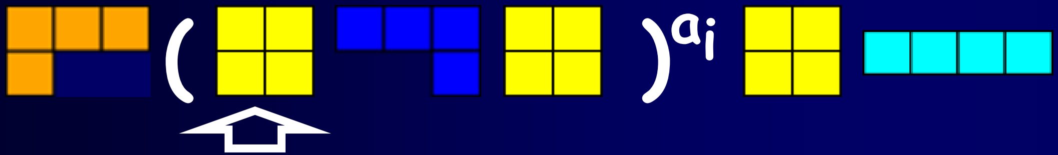




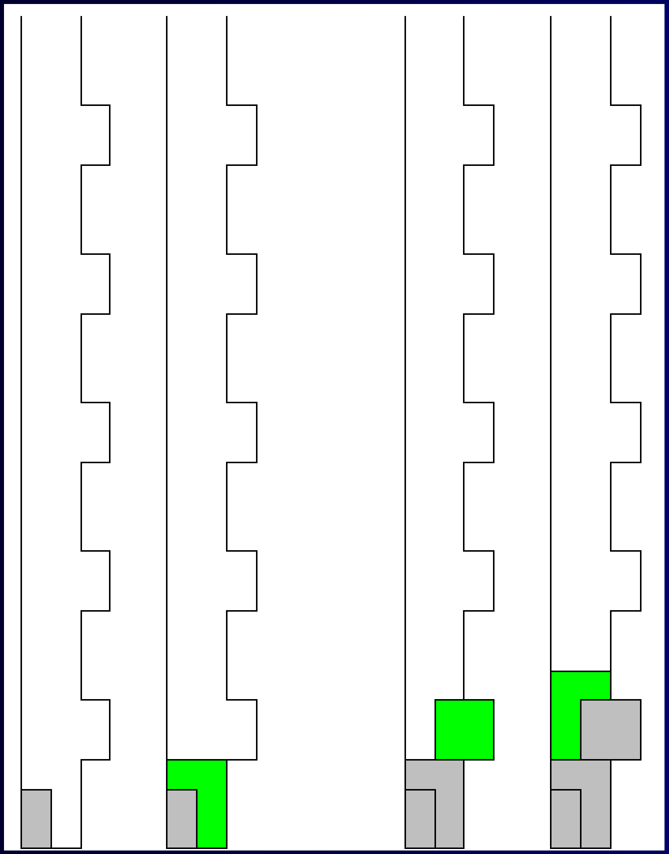
1

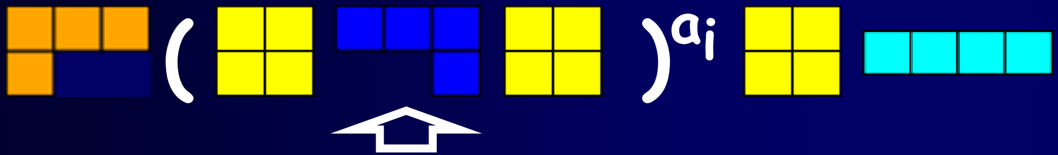
Les pièces qui correspondent à un a_i



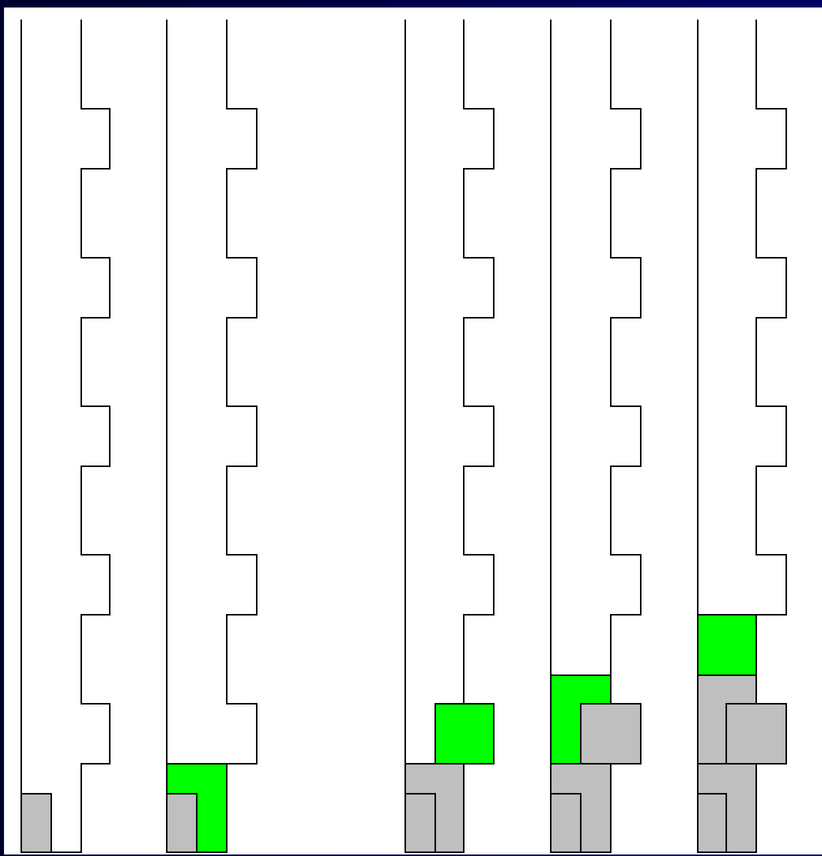


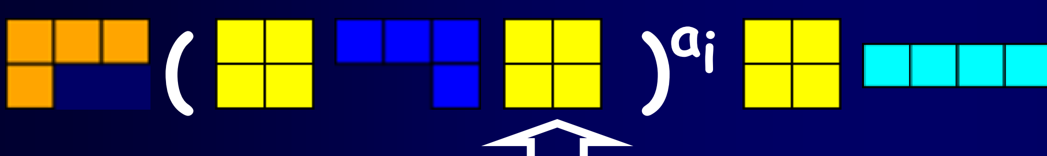
Les pièces qui correspondent à un a_i



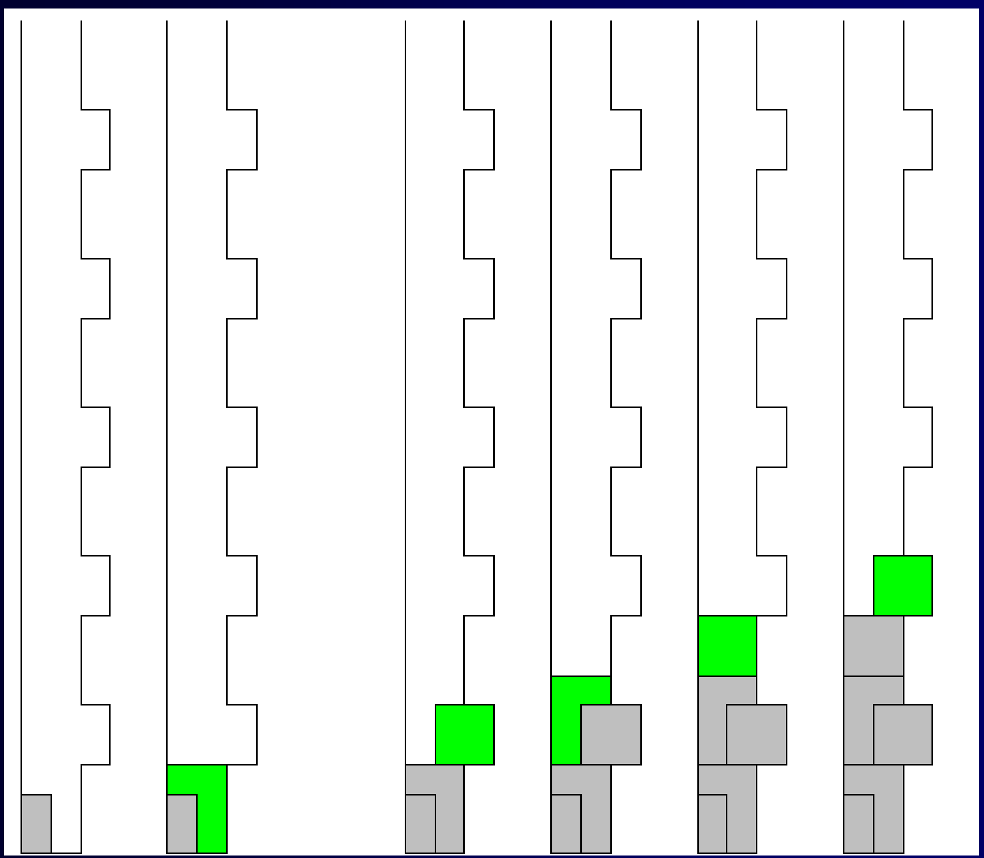


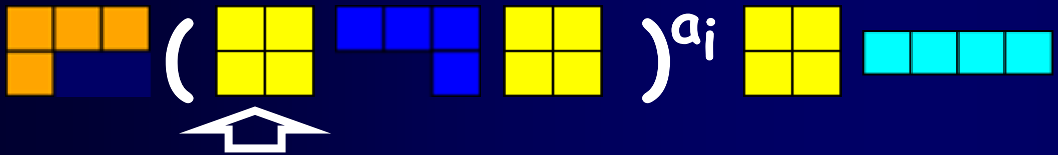
Les pièces qui correspondent à un a_i





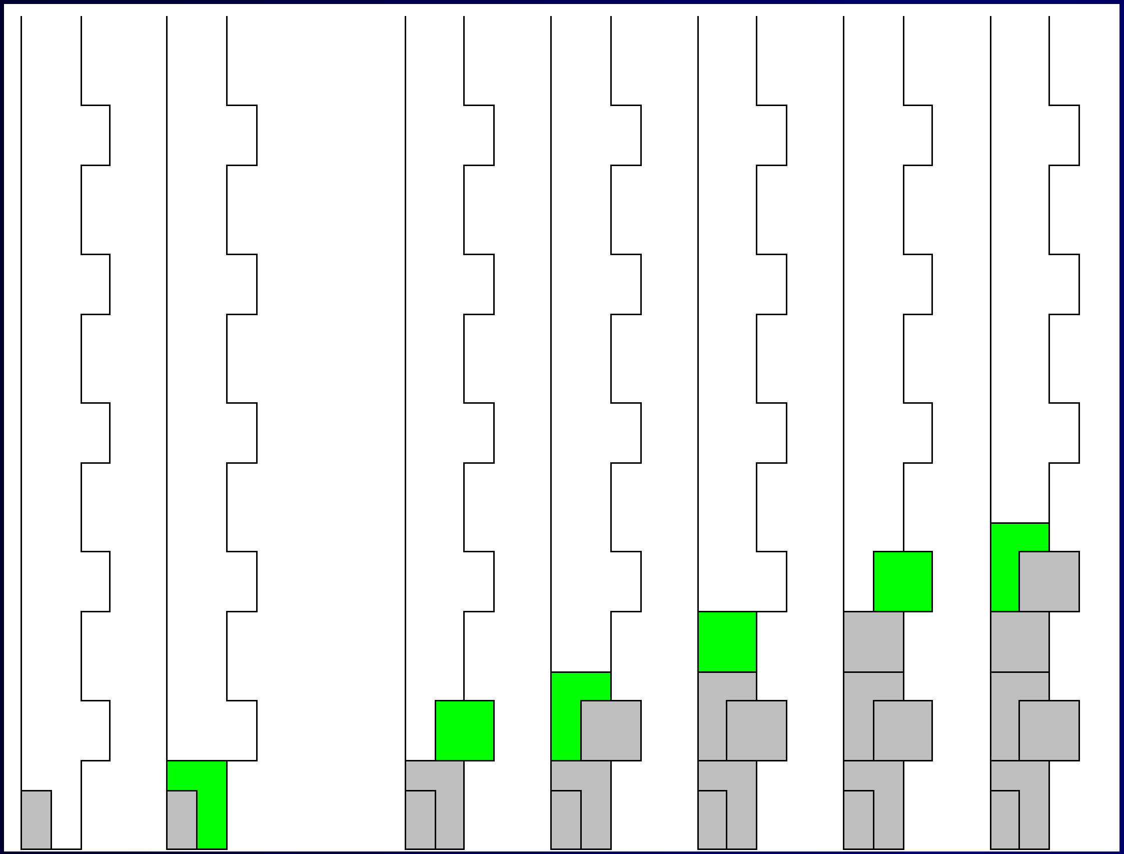
Les pièces qui correspondent à un a_i

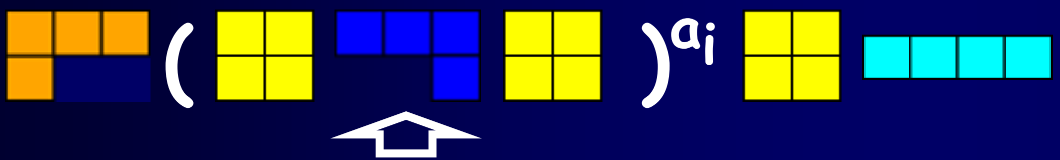




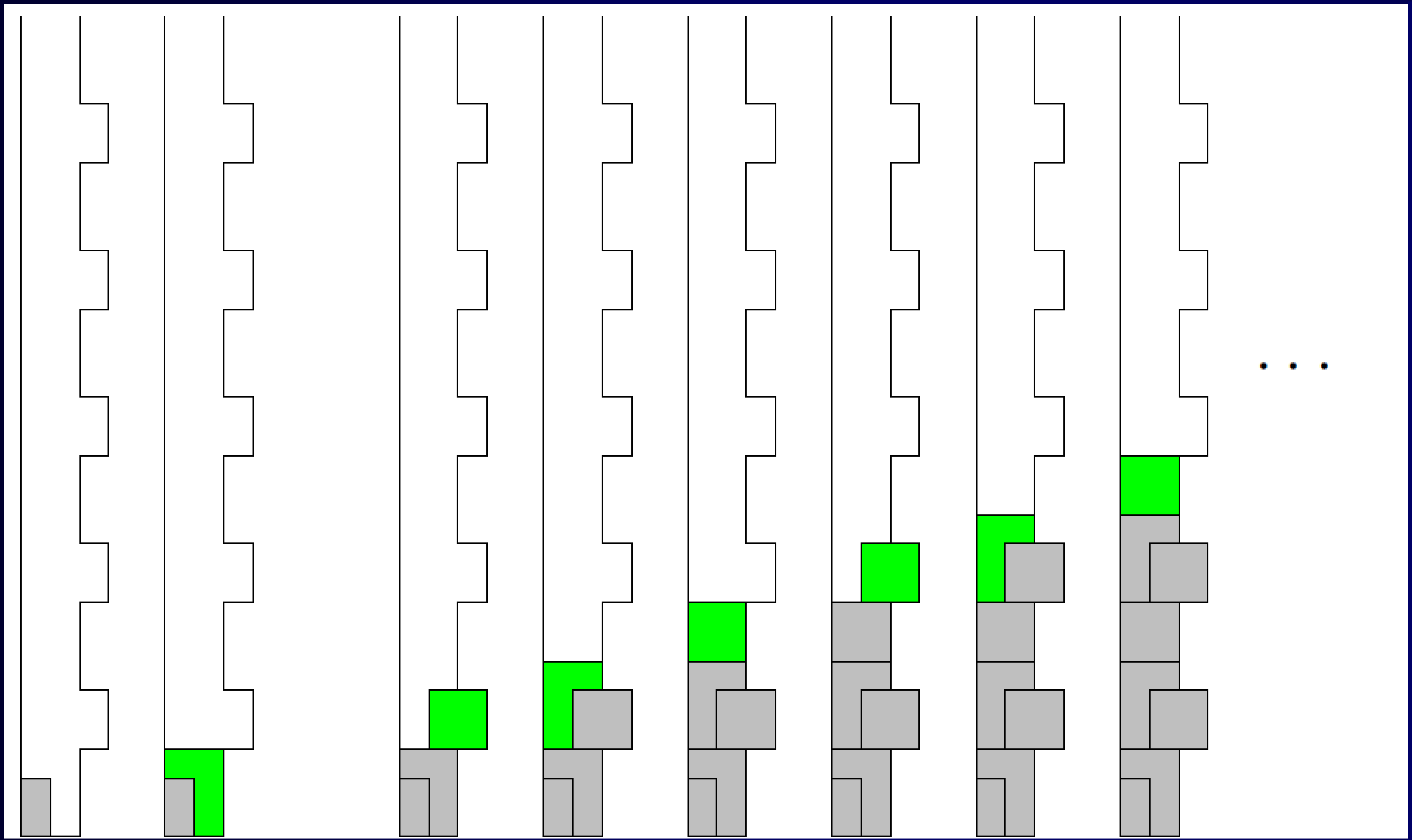
$(\quad)^{a_i}$

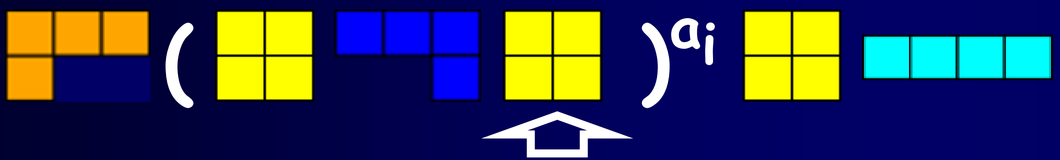
Les pièces qui correspondent à un a_i



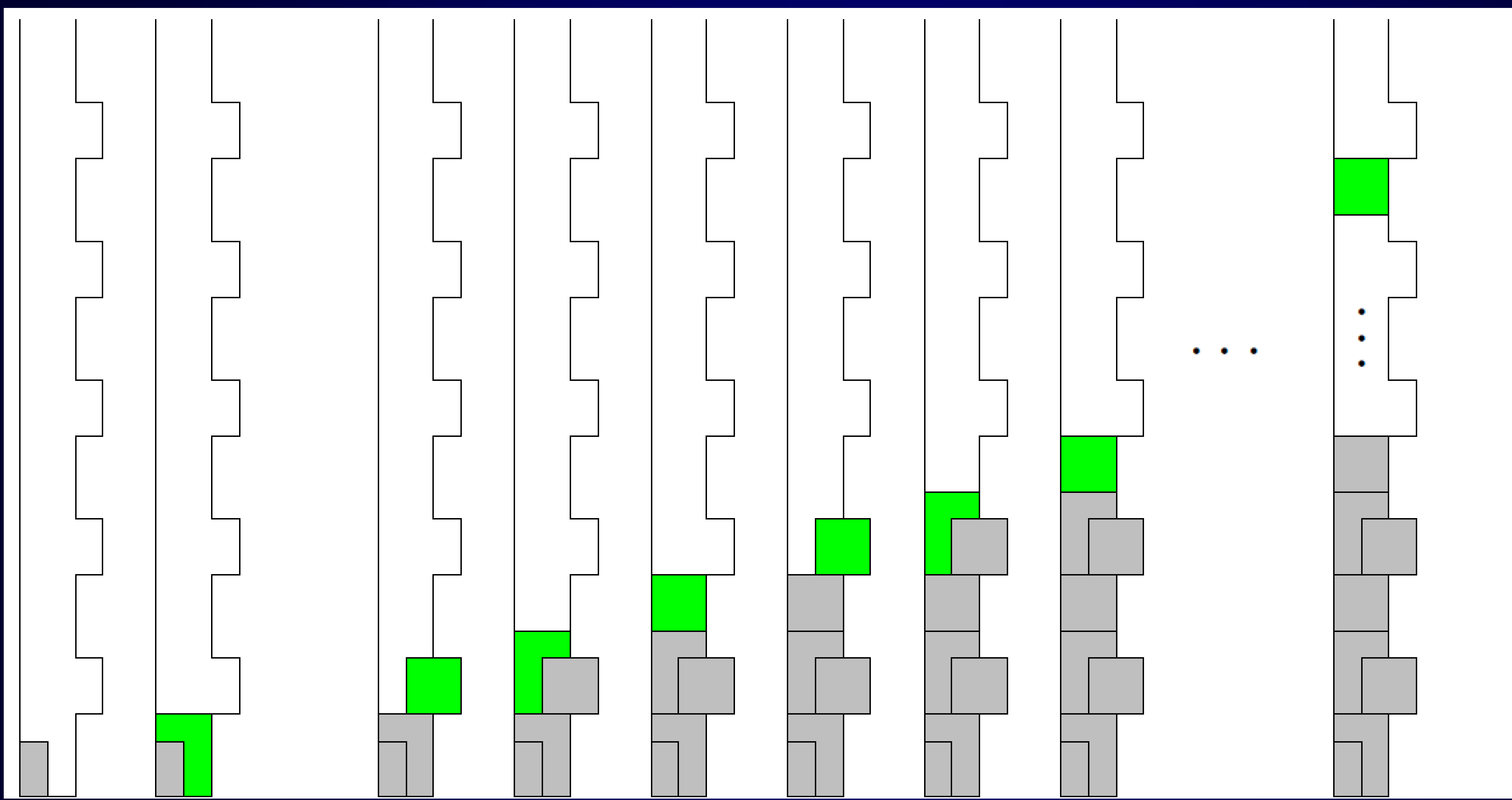


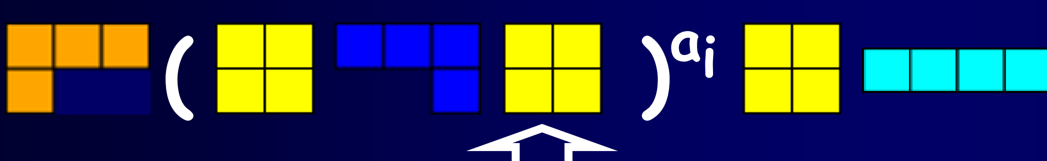
Les pièces qui correspondent à un a_i





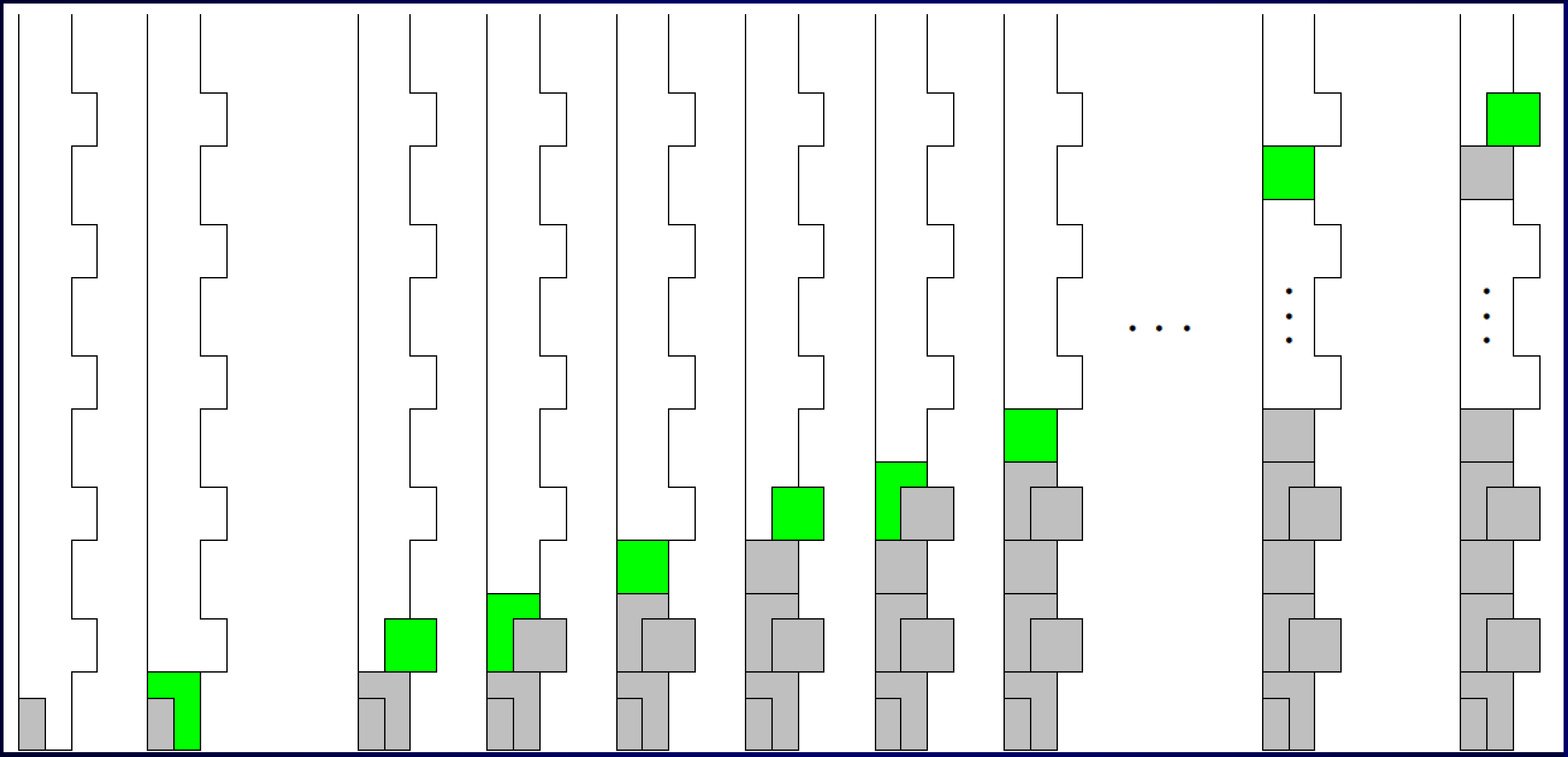
Les pièces qui correspondent à un a_i

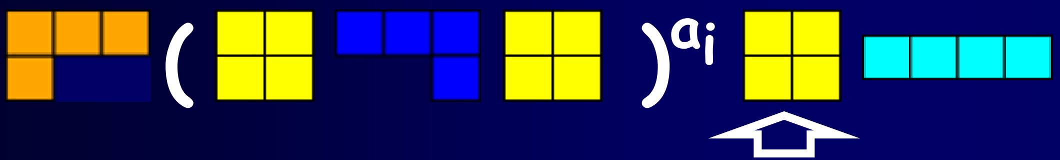




$1+3a_i$

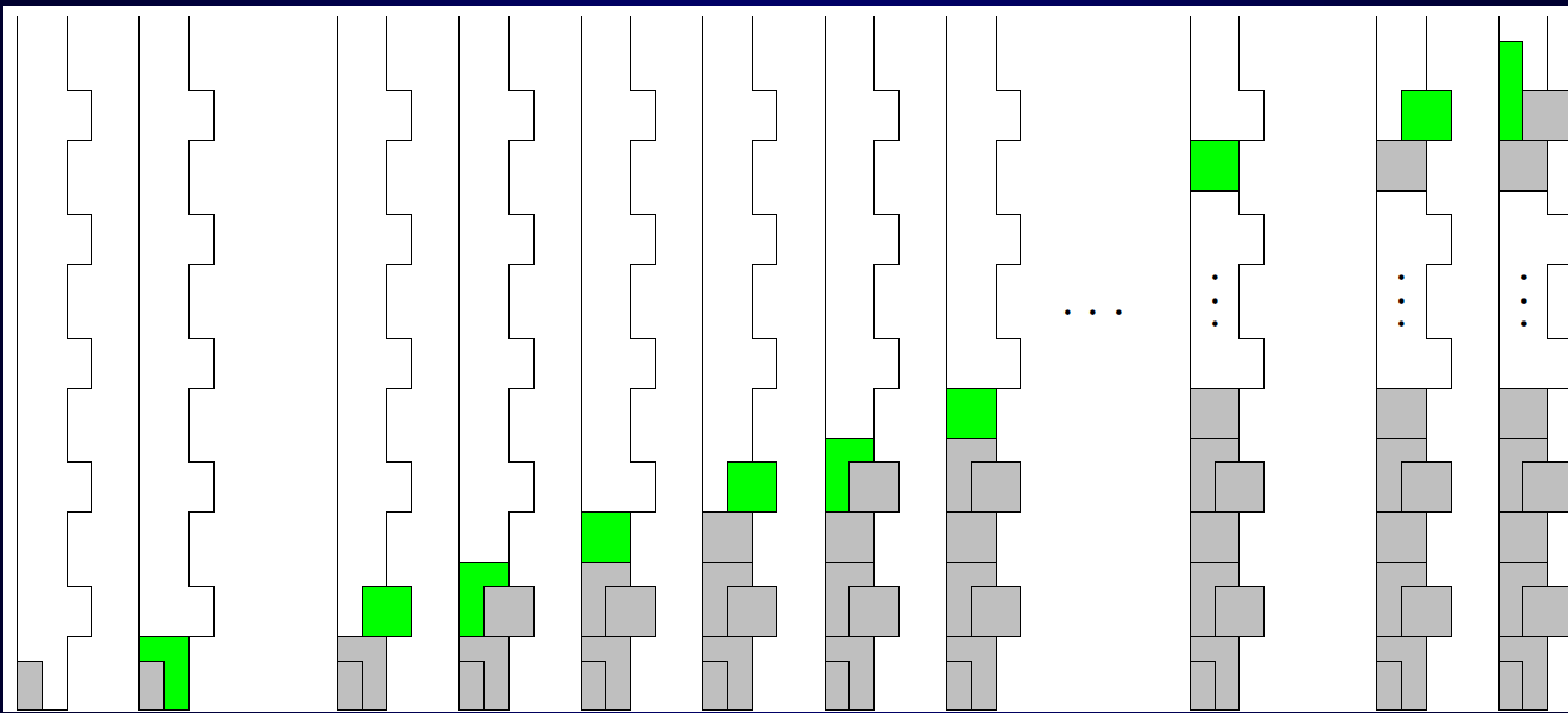
Les pièces qui correspondent à un a_i

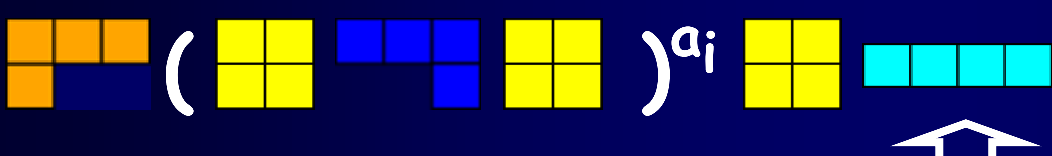




$1+3a_i+1$

Les pièces qui correspondent à un a_i





$1+3a_i+2$

OUI → OUI

Nous avons vu comment placer les pièces associées à un a_i.

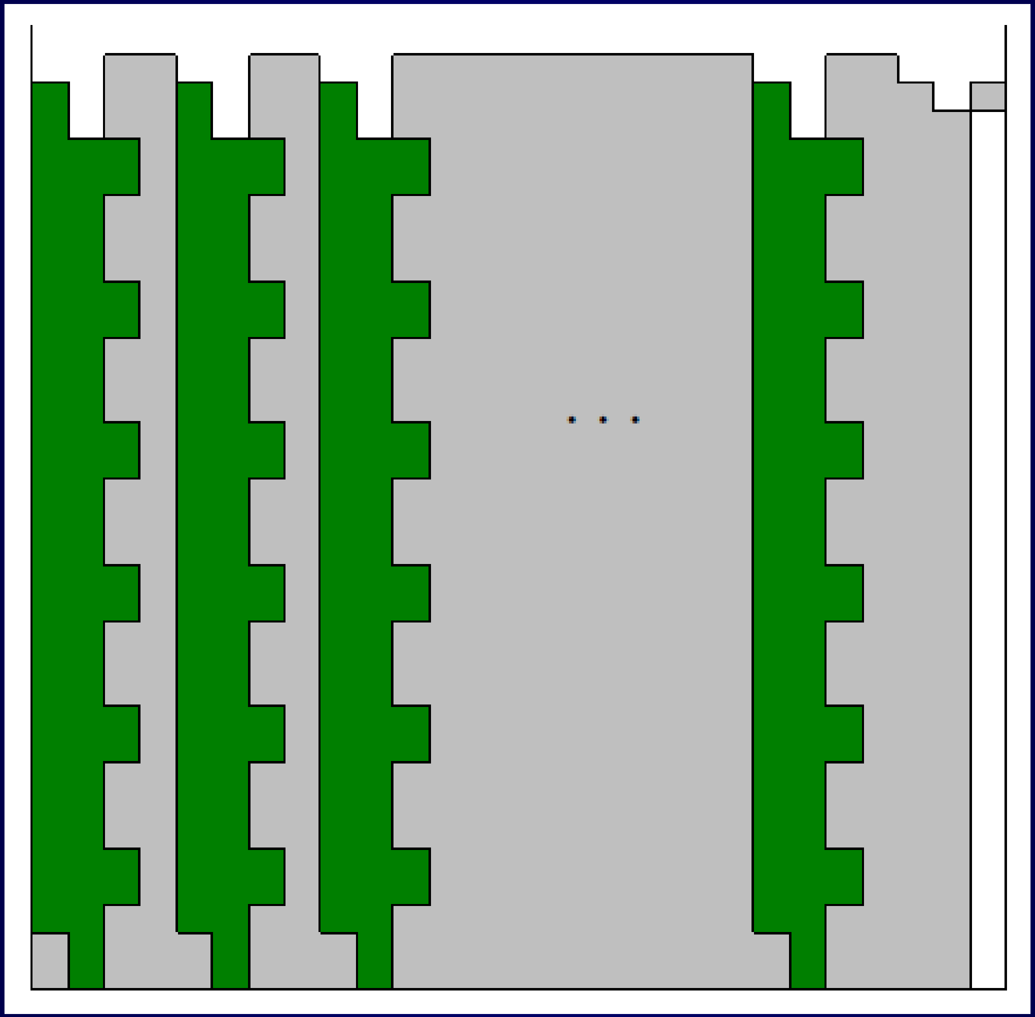
Ainsi, si on met toutes les pièces associées à un a_i, on peut continuer avec les pièces d'un a_j.

Les blocs d'un a_i permettent de remplir dans la *colonne* où il sont placés $3+5a_i+2=5(a_i+1)$ de hauteur.

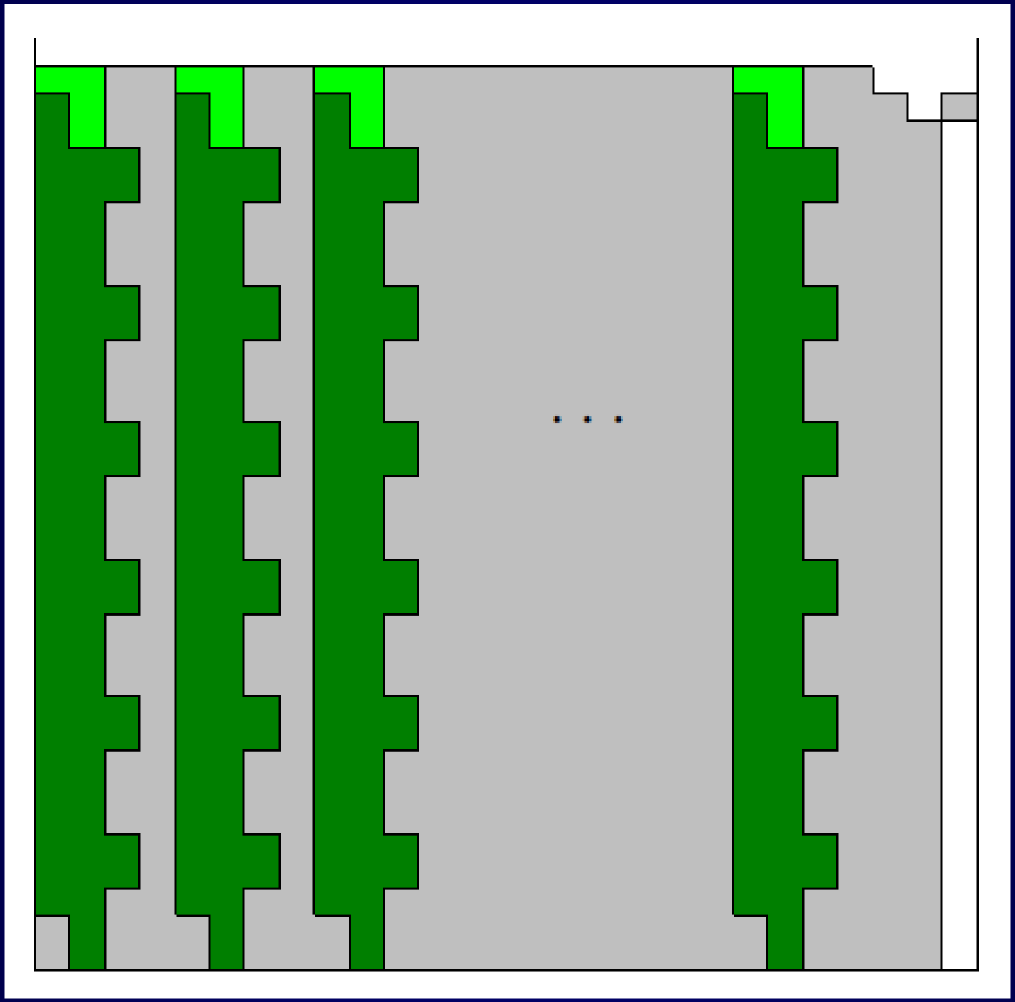
En plaçant les trois éléments d'un même ensemble A_i dans la *même colonne* on rempli donc 5T+15 lignes dans la *colonne*.

On fait cela avec toutes les pièces associés aux a_i.

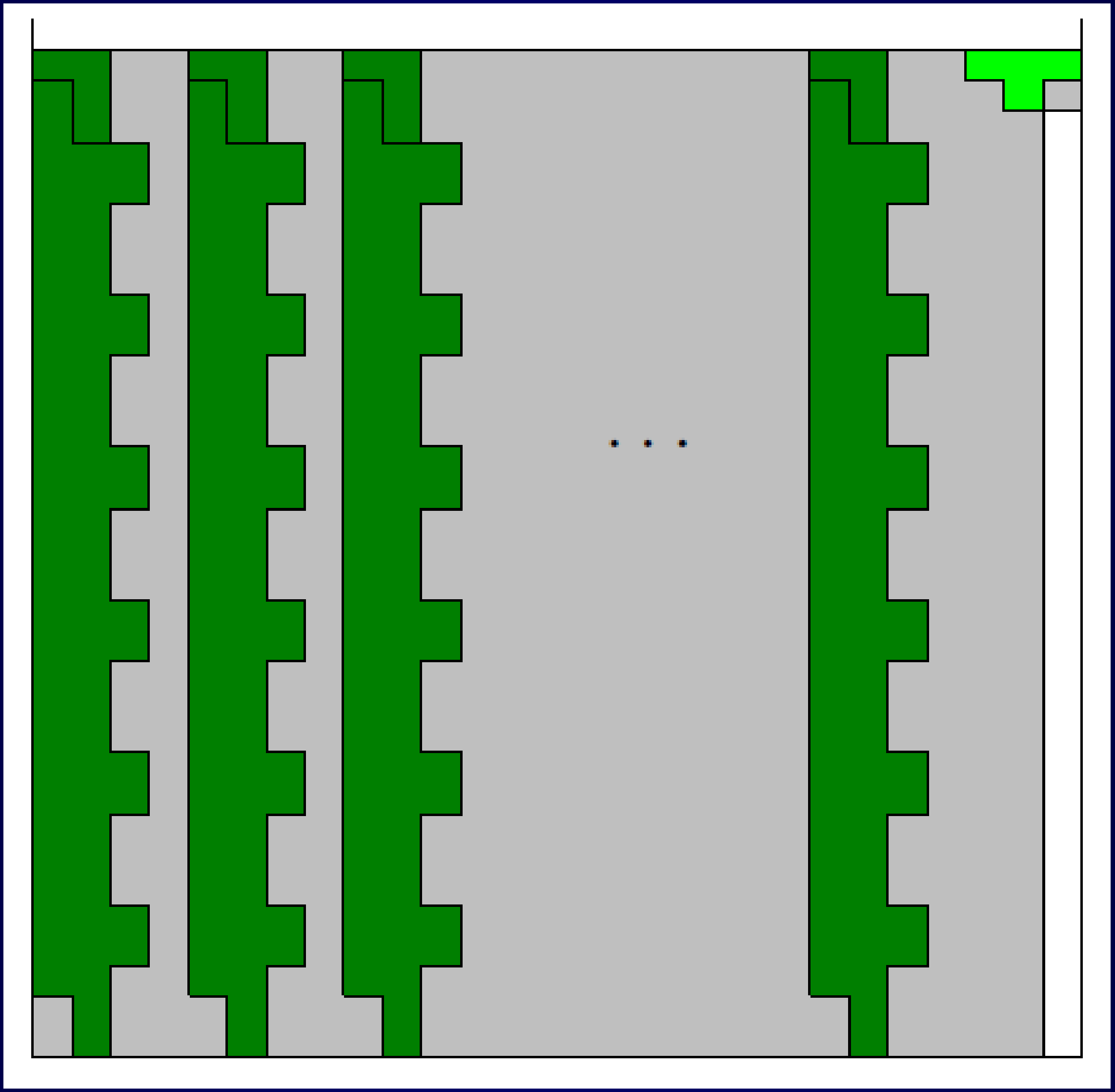
Situation après placement des a_i



Le placement des s pièces RG (hauteur 5T+18)

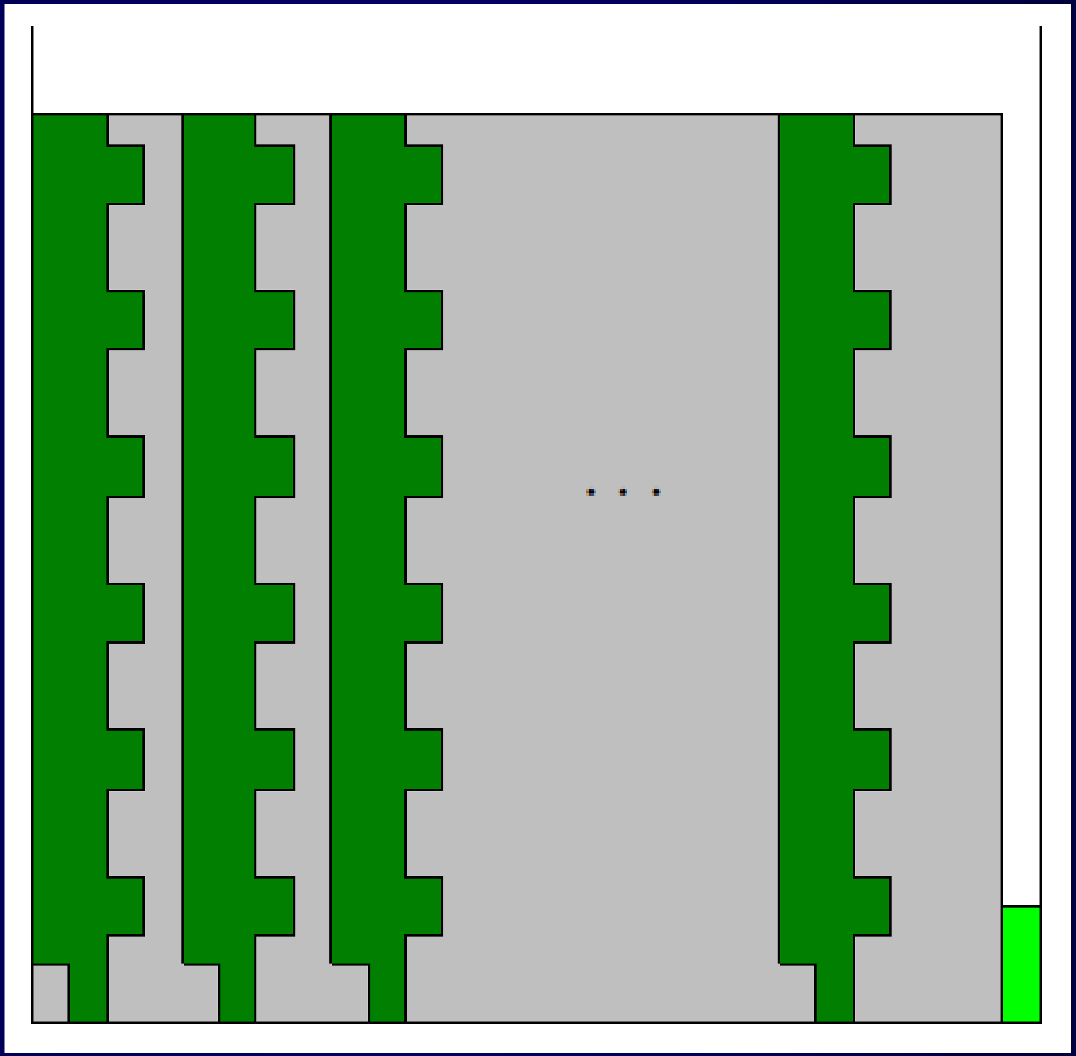


Le placement de la pièce T



Après suppression des deux lignes du haut

Hauteur 5T+16, qui sera comblée par les 5T/4+4 pièces de type I



▪ OUI ← OUI

- Comme on a vu, les 5T+18 lignes sont supprimées pour terminer avec un tableau vide.
- Si on a une solution, alors on ne peut à aucun moment poser aucun block qui dépasse des 5T+18 lignes. On justifie en comptant le nombre de carrés à remplir pour tout remplir jusqu'à la ligne 5T+18, qui est égale au nombre de carrés dans les blocks de notre liste.
- Le même comptage nous permet d'affirmer que les pièces avant le T permettent de remplir complètement les colonnes avec les vides. Et du coup, avant l'arrivée du T aucune ligne n'est supprimée.

- Ainsi si on a une solution qui permet d'arriver au tableau vide, c'est que les trois a_i qui remplissent chaque colonne sont de somme T.
- Donc il existe une triplets-Partition.

C.Q.F.D.

Et d'autres jeux ?

Echec	fini, mais en taille variable NP-difficile
Démineur	NP-complet (voir polycopié)
15-p	existence de la solution dans P, mais existence d'une solution en temps borné NP-complet
Go	NP-difficile
Otello	NP-difficile (taille variable)
Sokoban	NP-difficile
…	