

Examen 1 2018 Q7

On considère la machine de Turing dont la table de transition est :

	0	1	ε
1	(2, ε , +)	(3, ε , +)	(4, ε , -)
2	(2, 0, +)	(2, 1, +)	(4, 0, -)
3	(3, 1, +)	(3, 0, +)	(4, 0, +)

Quel est l'état de la bande lorsque la machine s'arrête, si elle a démarré dans l'état 1 avec sa tête de lecture positionnée comme suit :

...	ε	1	1	0	1	0	1	1	0	ε ...
		↑								

Examen 1 2018 Q8

Sachant que « 3-SAT » est le nom d'un problème de décision célèbre pour les informaticien(ne)s, connu pour être dans NP, que peut-on en dire ?

- ▶ « 3-SAT » n'a pas de solution (algorithme de résolution) :
vrai ? faux ? ne sait pas ?
- ▶ « 3-SAT » n'est pas dans P :
vrai ? faux ? ne sait pas ?
- ▶ On connaît des algorithmes efficaces pour résoudre toute instance de « 3-SAT » :
vrai ? faux ? ne sait pas ?
- ▶ Toute « solution » (instance positive) de « 3-SAT » est facilement vérifiable :
oui ? non ? ne sait pas ?

Examen 1 2018 Q9

Supposons que l'on sache qu'un problème de décision « X » n'est pas dans NP.
Que peut-on en dire ?

- ▶ « X » est dans P : vrai ? faux ? ne sait pas ?
- ▶ « X » est indécidable : oui ? non ? ne sait pas ?
- ▶ « X » est décidable et vérifier qu'une « solution » (instance positive) du problème « X » en est effectivement une prend un temps *au plus* polynomial par rapport à la taille de cette solution :
vrai ? faux ? ne sait pas ?
- ▶ Soit « X » est indécidable, soit vérifier qu'une « solution » (instance positive) du problème « X » en est effectivement une prend un temps *plus que* polynomial par rapport à la taille de cette solution :
vrai ? faux ? ne sait pas ?

Examen 1 2018 Q10

Supposons que l'on connaisse un algorithme de complexité $4n \log(n) + 3n + 2$ permettant de résoudre un problème de décision « Y » portant sur des données de taille n .
Que peut-on en déduire ?

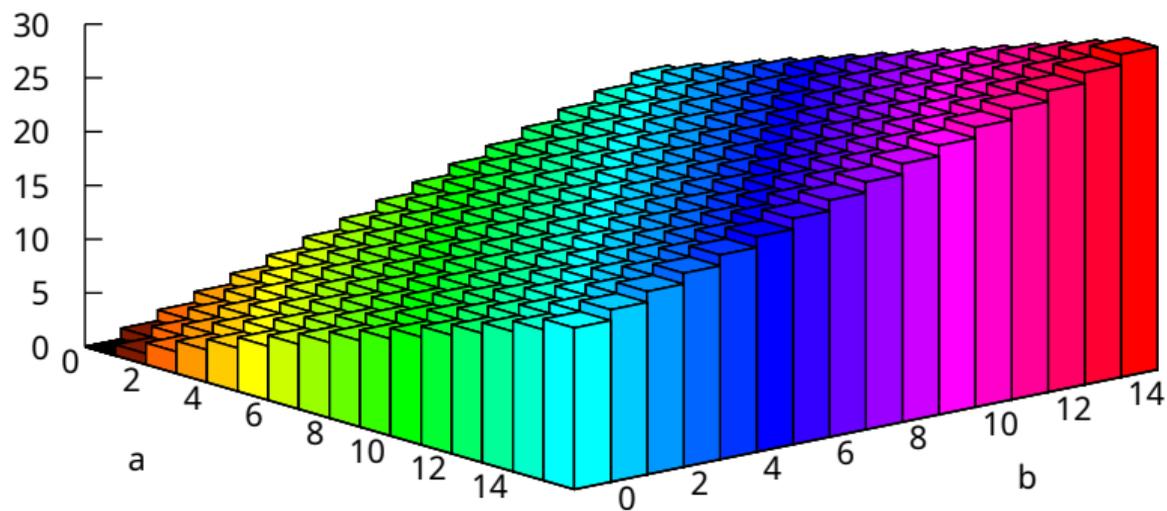
- ▶ « Y » n'est pas dans NP : vrai ? faux ? ne sait pas ?
- ▶ « Y » est dans NP, mais pas dans P : vrai ? faux ? ne sait pas ?
- ▶ « Y » est dans P : vrai ? faux ? ne sait pas ?

Leçon I.4 (Représentation de l'information) – Points clés

- ▶ nécessité d'une convention
on en a vu **trois** :
 - ▶ entiers positifs (ou nul) : `unsigned int`
 - ▶ entiers relatifs : `int` (complément à deux)
 - ▶ décimaux, représentation à virgule flottante : `double`
- ▶ nombre de bits pour représenter K objets
- ▶ domaine couvert
- ▶ précision (relative/absolue)
- ▶ comment utiliser les trois conventions ci-dessus

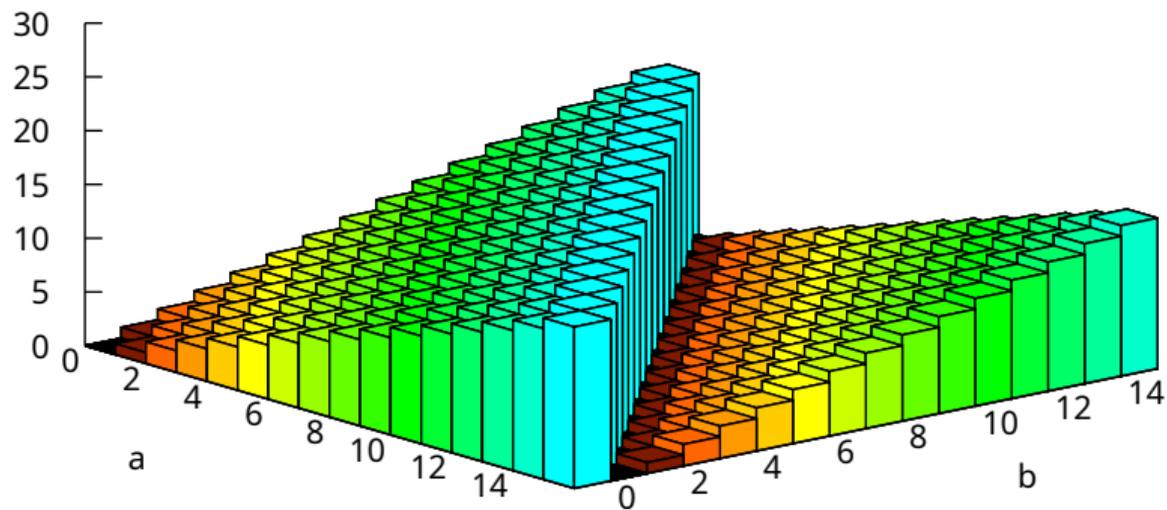
Leçon I.4 – Autre vue sur l'addition (sur 4 bits ici)

$\text{add}(a,b)$ en théorie



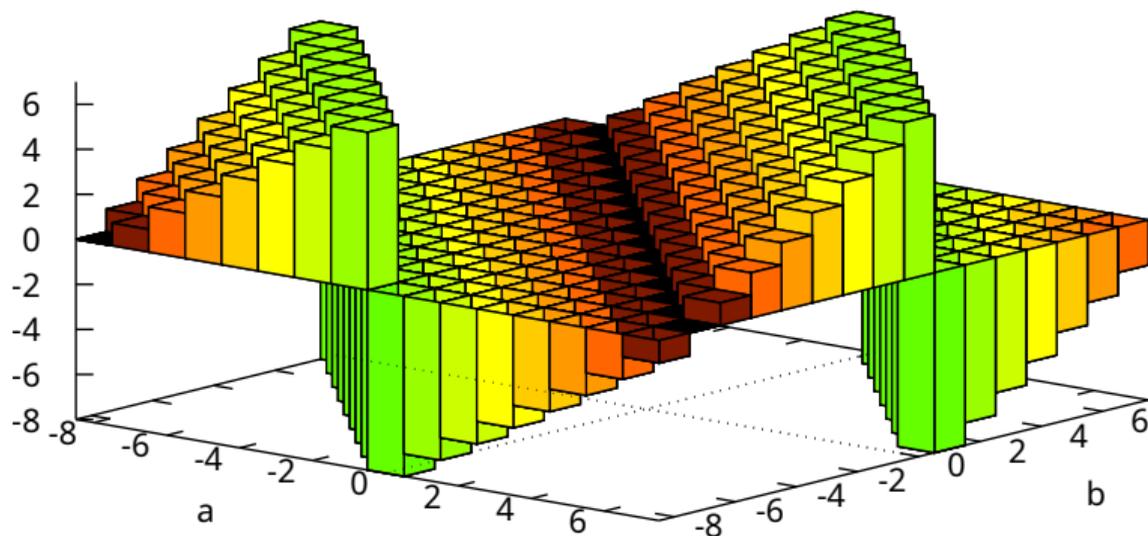
Leçon I.4 – Autre vue sur l'addition (sur 4 bits ici)

$\text{add}(a,b)$ sur 4 bits (non signé)



Leçon I.4 – Autre vue sur l'addition (sur 4 bits ici)

$\text{add}(a,b)$ sur 4 bits (signé complément à deux)



Leçon I.4 – Etude de cas 1

Que représente 10110101 ?

Leçon I.4 ? – Etude de cas 2

Donnez une version récursive de :

écriture en binaire

entrée : $n \in \mathbb{N}$

sortie : *écriture binaire de n*

$L \leftarrow ()$ // *liste vide*

Répéter

Si n est pair

| $L \leftarrow 0 \oplus L$ // *ajouter 0 devant*

Sinon

| $L \leftarrow 1 \oplus L$ // *ajouter 1 devant*

$n \leftarrow \lfloor \frac{n}{2} \rfloor$

Tant que $n > 0$

Sortir : L