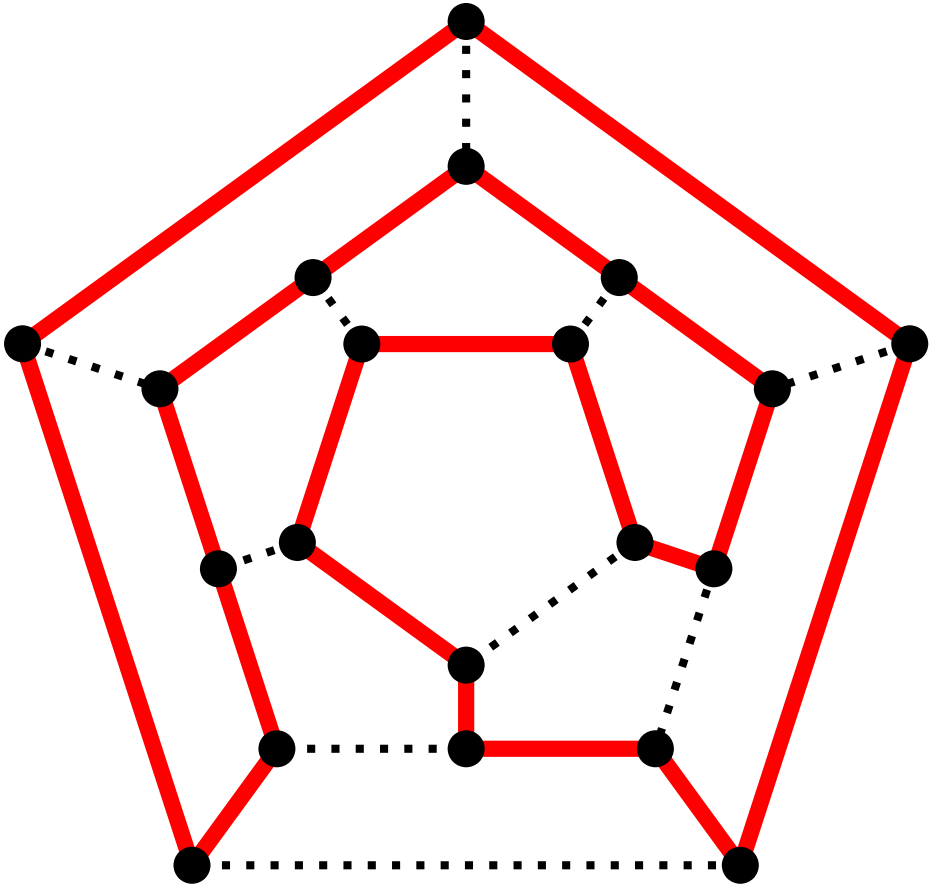
Les algorithmes



CS IRL

Computer Science In Real Life

À propos de ce document

- La page web du projet est ici : `http://www.loria.fr/~quinson/Mediation/CSIRL/`
- Les sources des ressources du projet sont entre autres ici : `http://github.com/jcb/CSIRL`
- Si vous le souhaitez, vous pouvez nous joindre ici : `discussions@listes.nybi.cc`

Crédits image

P1 : Chemin hamiltonien par Ch. Sommer, licence GFDL/CC-BY-SA
`http://en.wikipedia.org/wiki/File:Hamiltonian_path.svg`
P4 : Computer Science Major : CC-BY-NC `http://abstrusegoose.com/206`
P11 : Electric City par Mathias M, licence CC-BY-SA `http://www.flickr.com/photos/mathias_m/342535332/`

CS IRL ? Qu'est ce que c'est ?

- Des activités présentant les bases de l'informatique, mais sans ordinateur
- Pour chaque activité, un support matériel est proposé pour permettre d'*apprendre avec les mains*
- Les activités sont rangées en séances cohérentes et progressives

🔎 Computer Science In Real Life : Computer Science est la science informatique en anglais, tandis que IRL est l'abréviation utilisée sur internet pour décrire la vraie vie, ce qui n'est pas sur internet.

Les séances existantes dans la série

- Les algorithmes : Qu'est ce qu'un algorithme ? Et une heuristique ? À quoi ça sert ?
- Codes et représentations : Comment les ordinateurs codent et manipulent les données *(à venir)*
- Turzzle : puzzle de programmation sans ordinateur *(à venir)*

Objectif de la séance algorithmique

- Expliquer ce qu'est un algorithme et à quoi ça sert quand on veut utiliser un ordinateur
- Montrer un aspect du travail d'un informaticien, et de celui d'un chercheur en informatique
- La durée envisagée est d'une heure et demi ou deux heures.
- Ce n'est donc pas un cours complet sur l'algorithmique, qui nécessite 25 à 50h au minimum.
  Cours pour aller plus loin (en 48h) : `http://www.loria.fr/~quinson/Teaching/TOP/`

🔎 Si vous êtes l'animateur, vous trouverez des conseils et des astuces dans le coin de l'animateur en page 12.

Matériel nécessaire pour cette séance

- Des clous, dont un coloré
- Des petites planches de tailles différentes
- Des legos : cinq couleurs, avec à chaque fois deux pièces $\mathbf{2} \times \mathbf{2}$ et une pièce $\mathbf{4} \times \mathbf{2}$
- Une planche avec des clous plantés (mais qui dépassent) ; Une cordelette et un marqueur

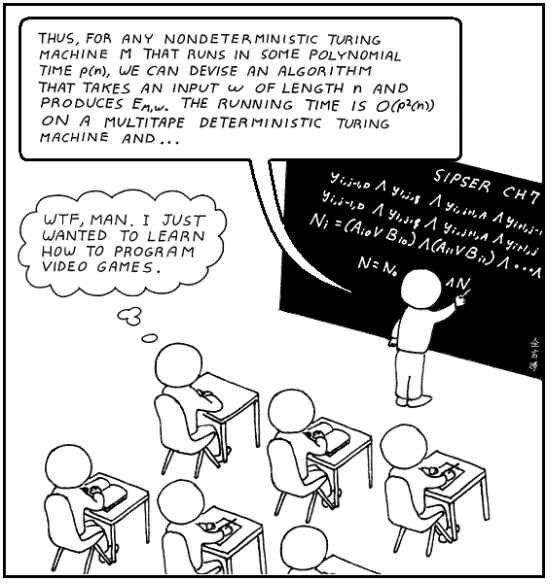
Computer Science IRL – La séance algorithmique

Introduction : les principales caractéristiques d'un ordinateur

- Il est très rapide : il peut calculer de 1 à 1 million en moins d'une seconde
- Il est parfaitement obéissant : il fait tout le temps exactement ce qu'on lui demande
- Il est absolument stupide : il exécute les ordres qu'on lui donne, sans la moindre capacité d'initiative.
  - Par exemple, si on demande à un ordinateur de s'arrêter, il le fait...
  - Autre exemple, quand j'indique à des amis comment venir chez moi, je leur donne des indications comme "troisième à droite" ou "à gauche au 2ieme feu". Si je me trompe dans mes indications ("à gauche" au lieu de "à droite") et que cela les ferait prendre l'autoroute à contre-sens, mes amis vont faire preuve de sens commun et ne pas appliquer la consigne. Les ordinateurs n'ont **aucun** sens commun.
  - Bug (n.m.) : consigne erronée donnée par un humain et appliquée bêtement par une machine.

Le travail d'un informaticien

- Se faire obéir d'un serviteur aussi stupide qu'un tas de fil demande un peu d'organisation
- Pour décomposer suffisamment les tâches à réaliser, il réfléchit à **comment** faire
  (un peu comme un cycliste qui descendrait du vélo pour se regarder pédaler afin d'expliquer ensuite comment faire)
- Pour chaque problème, il faut d'abord définir :
  - la situation initiale : le point de départ du problème
  - les opérations possibles : ce que j'ai le droit de faire pour faire évoluer la situation
  - la situation finale : ce vers quoi je veux tendre, l'état du problème quand je l'ai résolu



Activité : le jeu de Nim

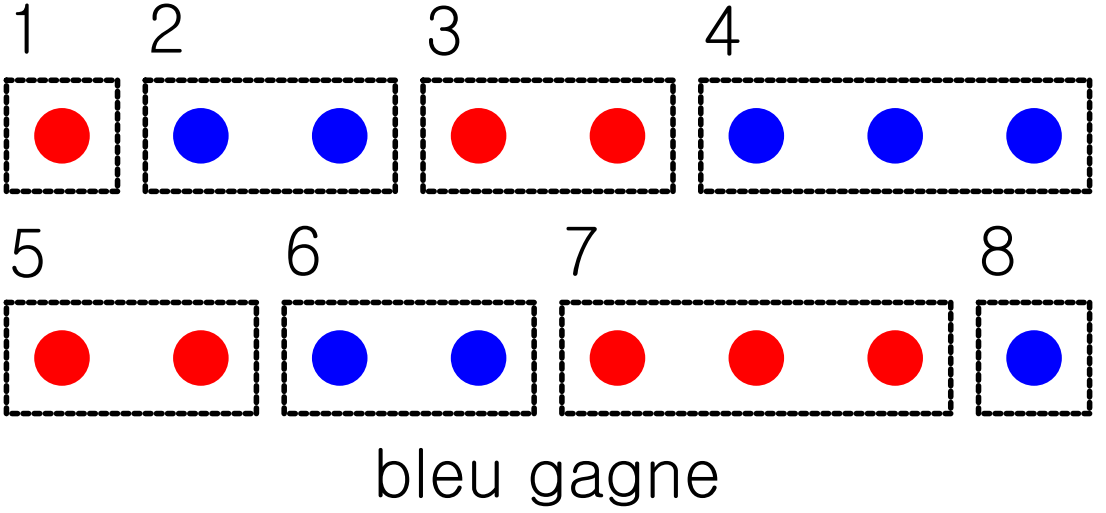
Voici un premier petit jeu simple, pour rentrer dans le sujet.

Matériel

- 16 petits objets (clous, allumettes, boulettes de papier ... peu importe !)

Règle du jeu

- Disposer les 16 objets sur une table
- Chaque joueur prend tour à tour 1, 2 ou 3 objets
- Celui qui prend le dernier objet à gagné



Ce qu'il faut retenir du jeu de Nim

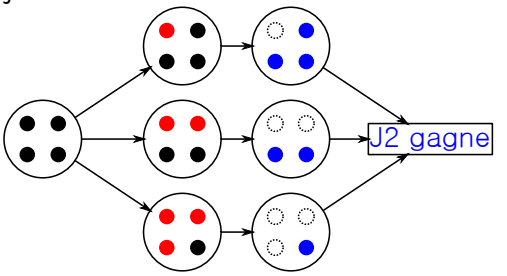
L'intérêt majeur de ce jeu est qu'il est sans suspense (voire, sans intérêt ;)

- Celui qui commence (J1) perd, car il existe un truc pour que J2 gagne à tous les coups
- Stratégie gagnante : Laisser 4, 8, 12 ou 16 objets à l'adversaire (un multiple de 4)

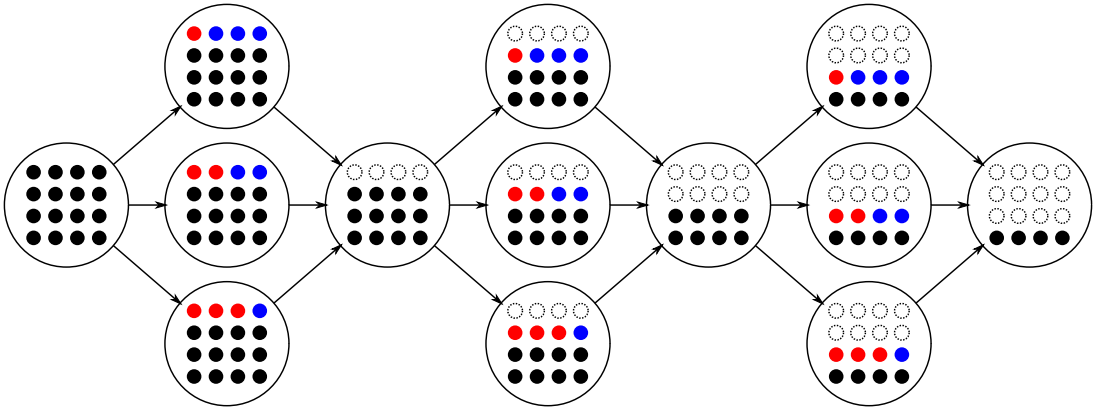
Se convaincre de l'efficacité de la stratégie gagnante

Prenons le dernier tour comme exemple. Il reste 4 objets, et J1 joue.

- Si J1 prend 1 objet, J2 en prend 3 (dont le dernier)
- Si J1 prend 2 objets, J2 en prend 2 (dont le dernier)
- Si J1 prend 3 objets, J2 en prend 1 (le dernier)



Quoi qu'il fasse, J1 a donc perdu dans ce cas, si J2 sait jouer.
En appliquant la même méthode, J2 peut guider le jeu de manière à passer de 16 objets à 12, puis 8 et enfin 4.
Donc J1 a perdu avant de commencer.



Le rapport avec l'informatique

- Passer de la situation initiale à la situation finale à coup sûr demande d'avoir une *stratégie gagnante*
- C'est un **algorithme** en informatique, une recette de cuisine ou un manuel de montage de meubles
- Pour se faire obéir du tas de fils, l'informaticien cherche l'algorithme pour résoudre le problème, puis il écrit le **programme** (traduction de l'algorithme dans un langage informatique)

pour aller plus loin ...

- Sous quelle condition est-on sûr de gagner si le nombre de objets n'est pas un multiple de 4 ?
- Que faudrait-il changer pour gagner à coup sûr si les joueurs ne peuvent prendre qu'un ou deux objets à la fois ?

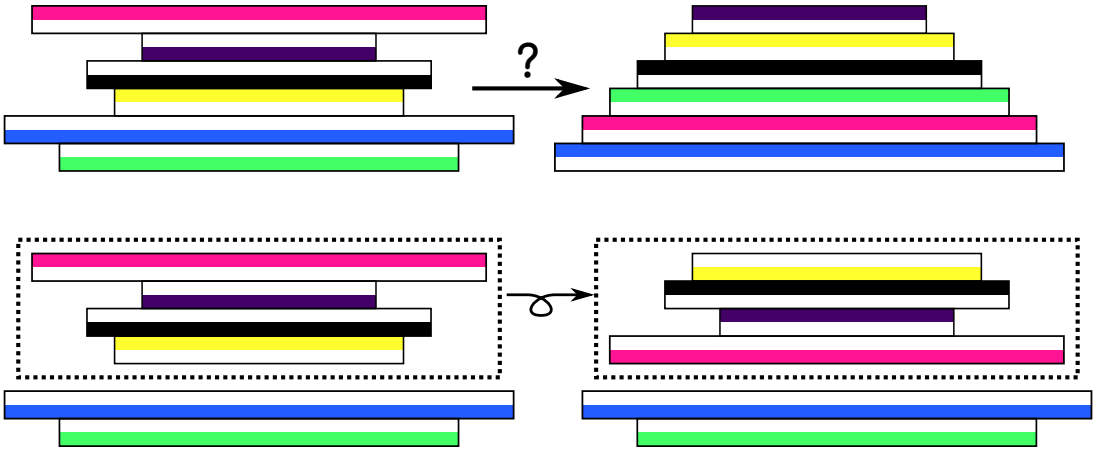
Activité : Le crêpier psycho-rigide

Matériel

- des planchettes en bois de tailles et de couleurs différentes (faces reconnaissables)
- éventuellement une pelle à tarte pour retourner les planchettes

Règle du jeu

- Objectif : ranger les crêpes de la plus grande (en dessous) à la plus petite (au dessus), face colorée vers le haut.
- La seule action possible est de prendre une ou plusieurs crêpes sur le haut de la pile, et de les reposer à l'envers.



Ce qu'il faut retenir du crêpier psycho-rigide

Un algorithme

- n'a d'intérêt que si on peut l'expliquer
- doit être suffisamment simple pour pouvoir l'expliquer à une machine
- «**Diviser pour mieux régner**» : on essaie toujours de décomposer un algorithme en tâches simples

L'algorithme que doit suivre le crêpier est :

- ramener la plus grande crêpe en haut de la pile
- retourner pour que la face brûlée soit vers le haut
- retourner la pile de sorte à mettre la plus grande crêpe en bas
- réitérer avec la crêpe de taille inférieure

Le rapport avec l'informatique

- l'informaticien passe son temps à trouver des algorithmes et à les expliquer à la machine
- le principe «**Diviser pour mieux régner**» est fondamental en informatique

Activité : le plus court chemin

Matériel nécessaire

- Une planche avec des trous au hasard
- Autant de longs clous que de trous
- Une ficelle suffisamment longue
- Un marqueur

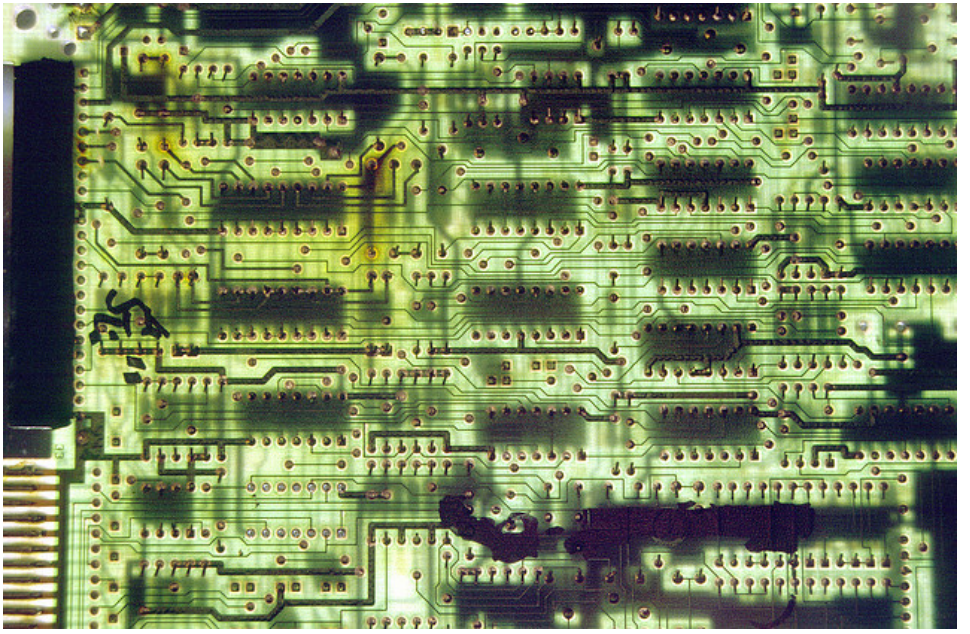
Règles du jeu

- Situation initiale : les clous sont mis dans les trous et leurs têtes dépassent de la planche
- Coups autorisés :
  - On part d'un clou au hasard et on passe la ficelle de clou en clou
  - On doit passer par tous les clous
  - On ne peut passer qu'une seule fois par le même clou
  - On doit revenir au point de départ
- Situation finale : L'objectif est de passer par tous les clous en faisant le plus court chemin. À chaque fois qu'un record est battu, on fait une marque sur la ficelle.

Objectif de l'activité

- Certains algorithmes sont trop compliqués pour trouver la solution optimale en un temps raisonnable. Parfois, il vaut mieux trouver une solution approchée très rapidement, c'est une heuristique
- algos NP, NP-difficiles, NP-complets
- C'est un problème qui parait bête mais qui est complexe et qui a de nombreuses applications dans la vie réelle
- Exemple d'application amusante de notion de NP-complétude : `http://arxiv.org/abs/1203.1895`

Ce qu'il faut retenir du plus court chemin



Le coin de l'animateur

Trucs et astuces pour s'assurer que le message passe bien

Remarques générales

- Il faut vous approprier les activités. N'hésitez pas à ne pas suivre les consignes à la lettre.
  Ces activités sont des bases de discussion avec les participants, il n'y a pas d'évaluation à la fin.
- Une question récurrente des participants est de savoir ce que l'animateur fait, en recherche.
  Pensez à préparer une présentation compréhensible de vos recherches, avec le domaine général, ses difficultés et applications et quelques mots de vos préoccupations propres.
  Exemple : Le domaine de mon travail est le parallélisme : est ce que ranger sa chambre va plus vite à 2 ou 3 ? oui. à 3000 ? Non, on perd du temps à se coordonner. Et pourtant, la météo de demain est calculée en utilisant plusieurs milliers d'ordis en même temps, ce qui est difficile. Dans ce domaine, mon travail à moi est d'établir des instruments scientifiques (simulateurs ou parcs de machines), comparables aux télescopes ou microscopes des physiciens, et qui servent d'outils aux scientifiques du domaine.

À propos du mot d'introduction

- On peut faire cette présentation soit au début, soit juste après l'activité sur le jeu de Nim.
- Commencer directement par un petit jeu permet d'éviter que les participants ne décrochent avant même qu'on ne commence.

À propos du jeu de Nim

- L'objectif de cette activité est simplement d'introduire la notion d'algorithme
- On propose le jeu avec le participant, mais sans dire trop vite qu'on a un truc. S'il y a plusieurs participants, on jouera avec plusieurs personnes, pour laisser sa chance à chacun. On peut faire une sorte de petit tournois.
- Il faut bien sûr laisser commencer le participant pour gagner à coup sûr. S'il insiste pour ne pas commencer, on peut le faire (et rattraper la stratégie gagnante à la première erreur du participant)
- On n'introduit l'existence du truc pour gagner que plus tard, quand on gagne à plate couture
- Si on perd, c'est à dire si on n'a pas réussi à appliquer la stratégie gagnante, il faut proposer un match en 3 (ou en 5 en cas de coup dur ;)
- On peut amener le participant à découvrir la stratégie gagnante en groupant les clous par paquets de 4 au lieu de la disposition pyramidale.
- Si l'un des participants connaît déjà la stratégie gagnante du jeu, il peut remplacer l'animateur dans une partie avec d'autres participants

Le coin de l'animateur

Trucs et astuces pour s'assurer que le message passe bien

À propos du jeu du crêpier psycho-rigide

- L'objectif de cette activité est de trouver un algorithme et de le faire verbaliser par les participants
- On propose au participant de d'abord tenter de le résoudre intuitivement, sans réfléchir
- Si le participant bloque, on peut lui donner un conseil : « Une bonne première étape est de se débrouiller pour mettre la grande en bas »
- Si le participant bloque toujours, on peut lui donner un second conseil : « où est-ce que la grande devrait être pour pouvoir la mettre en bas ? »puis le guider pour l'étape suivante.
- On essaie ensuite de faire expliquer l'algorithme par le participant. On gagne à ce que ce soit le participant et non l'animateur qui explique aux autres, avec ses propres mots.

Le coin de l'animateur

Trucs et astuces pour s'assurer que le message passe bien

À propos du base-ball multicolore

- L'objectif de cette activité est d'introduire les notions de correction et performances d'algorithmes
- Il faut laisser les participants chercher un peu en les faisant verbaliser
- S'ils sont sur le point de trouver l'algo juste, on introduit très vite l'algo faux pour préserver un enchaînement logique : "oui, ok, mais je vais vous montrer une façon de faire rigolote"
- Quand l'algo juste est établi, et avant de parler de performance, on peut appliquer sur une variante :
  - Chaque participant prend une couleur (une maison pour commencer) et ses pieds
  - Chaque participant prend une couleur (une maison pour commencer) et ses pieds
  - Chaque participant (sauf 1) prend un bonhomme dans chaque main
  - À chaque étape, celui qui a une main libre prend un bonhomme dans la main d'un voisin
  - (attention, c'est fastidieux à 8 ou 9 couleurs, il vaut mieux faire deux rondes car l'algo semble $\mathbf{O(n^2)}$)
- Expérimentalement, l'algo qui tourne converge très souvent vers la solution à 5 maisons, mais converge souvent vers la boucle infinie quand il y a plus de couleurs. Ne tentez pas le diable ;)
- Dans la disposition linéaire, il est plus simple de mettre la couleur avec un seul bonhomme à une extrémité, et commencer par remplir la maison de l'autre extrémité. Sinon, on se retrouve avec une maison remplie de un seul au milieu, et il faut comprendre que la solution passe par le stockage temporaire d'un pion de la maison d'à coté sur le trou.
- Le discours sur le $\mathbf{O(n)}$ est volontairement approximatif. On veut faire sentir les choses ; faire un vrai cours prend une douzaine d'heures (cf. `http://www.loria.fr/~quinson/Teaching/TOP/`).
- Il serait intéressant de prouver effectivement la correction de l'algorithme linéaire, ainsi que de quantifier la probabilité de fonctionner de l'algo qui tourne en fonction du nombre de maisons
- Au passage, le crépier ne ressemble pas du tout aux tours de Hanoï : l'histoire ressemble un peu, mais la résolution est très différente (il y a $\mathbf{2^n - 1}$ étapes à Hanoï et $\mathbf{3 \times n}$ au crépier...)