

# Algorithmes et complexité (2022-2023)

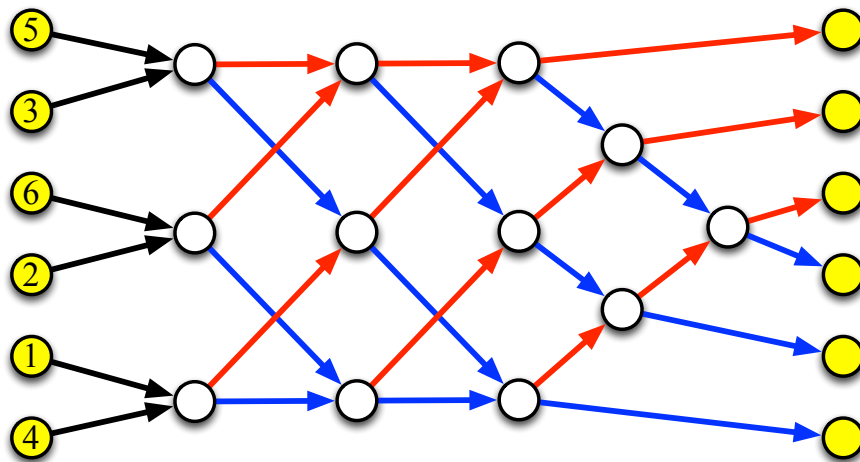
Florian Bridoux, François Doré, Dorian Mazauric

## Travaux Dirigés 1

### Exercice 1 :

Expérimenter le jeu du réseau de tri à 6 éléments suivants.

Chaque rond blanc représente une instruction de l'algorithme : comparaison des deux nombres, puis le plus petit suit la flèche rouge, le plus grand la flèche bleue.



Construire un réseau de tri à 3 éléments ? à 4 éléments ?  
Prouver vos algorithmes ?

### Exercice 2 :

Un apprenti cuisinier a préparé (et réussi) des crêpes, mais les a empilées n'importe comment. Son chef, un peu maniaque, veut les ranger de la plus grande (en bas) à la plus petite (en haut).

Pour cela, il dispose d'une spatule qu'il peut glisser entre deux crêpes pour retourner complètement le haut de la pile. Le chef étant également une personne pressée, veut réordonner les crêpes le plus rapidement possible, c'est-à-dire en utilisant le moins de mouvements possible (un mouvement = glissement de spatule entre deux crêpes et retournement du haut de la pile).

Décrire informellement un algorithme pour résoudre ce problème puis en pseudo-code.  
Quel est le nombre d'opérations dans le pire des cas (en fonction du nombre de crêpes) ?

### Exercice 3 :



## Jeu des bâtonnets

### Règle du jeu

Prenons quelques bâtonnets (allumettes, carottes...).

Deux joueurs prennent tour à tour **1, 2 ou 3** bâtonnets.



Le joueur qui prend le(s) dernier(s) bâtonnet(s) a **GAGNÉ** !

### Exemple de déroulement du jeu (11 bâtonnets)



Le **premier** joueur prend 2 bâtonnets et rend les 9 restants au **second** joueur.



Le **2nd** joueur prend 1 bâtonnet et rend les 8 restants au **1er** joueur.



Le **1er** joueur prend 2 bâtonnets et rend les 6 restants au **2nd** joueur.



Le **2nd** joueur prend 2 bâtonnets et rend les 4 restants au **1er** joueur.



Le **1er** joueur prend 1 bâtonnets et rend les 3 restants au **2nd** joueur.



Le **second** joueur prend les 3 derniers bâtonnets et **GAGNE** !

Le premier joueur a fait une GROSSE bêtise, savez-vous pourquoi ?

### Questions

Si le nombre initial de bâtonnets est 1, 2, 3, 4, 5, 6, 7, 8, 9..., qui du **premier** ou du **second** joueur peut gagner ?

Que se passe-t-il en modifiant un peu les règles ?

1. Si celui qui prend le(s) **dernier(s)** bâtonnet(s) **PERD** ?
2. Si on peut prendre **1, 2, 3 ou 4** bâtonnets à son tour ?
3. Si on peut prendre **jusqu'à 42** bâtonnets à son tour ?

Écrire un algorithme qui gagne à coup sûr (quand cela est possible) en pseudo-code ?

## Exercice 4 :



# Jeu de la tablette de chocolat

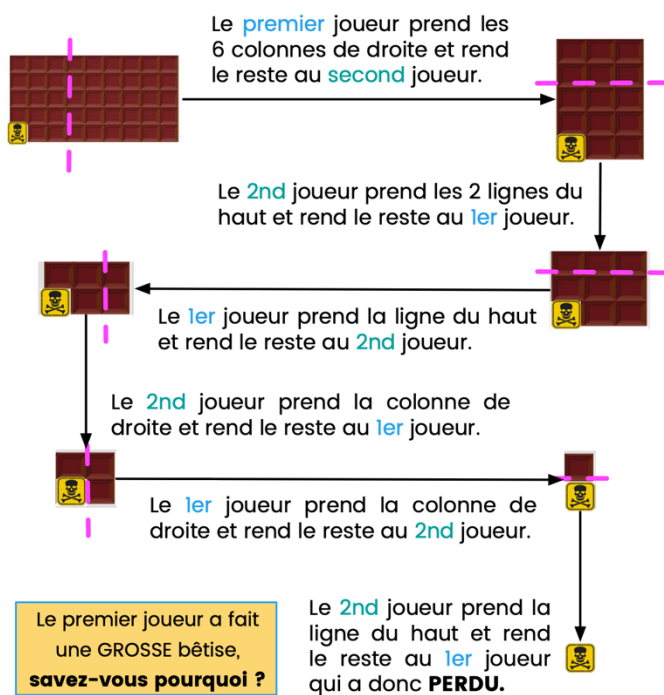
## Règle du jeu

Prenons une tablette de chocolat dont un carreau (celui en bas à gauche) est empoisonné. Deux joueurs cassent tour à tour la tablette (**en suivant une ligne ou une colonne**) et rendent un des 2 morceaux à l'adversaire



Le joueur qui se retrouve avec (uniquement) le carré empoisonné a **PERDU !**

## Exemple de déroulement du jeu (tablette 5x9)



## Comment gagner à coup sûr ?

Trouver un algorithme (recette, mode d'emploi, méthode) pour gagner à tous les coups...

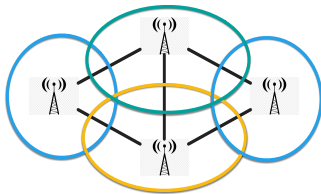
Écrire un algorithme qui gagne à coup sûr (quand cela est possible) en pseudo-code ?

## Exercice 5 :



# Coloration de graphes

## Allocation de fréquences



### Modélisation par un graphe :

**Sommet** = antenne

**Arête** entre deux antennes dont les zones d'émission s'intersectent.

**Couleur** = fréquence



Interférence dans les endroits où on reçoit deux antennes sur la même fréquence.

### Coloration propre :

sommets reliés par une arête ont des couleurs différentes.

### OBJECTIF

**minimiser** le nombre total de **fréquences**.  
(une bande de fréquences coûte cher)

**minimiser** le nombre total de **couleurs** dans une coloration propre

### JEU

Étant donné un graphe et un ensemble de couleurs, trouver une coloration propre du graphe ou prouver qu'il n'en existe pas.

## Un problème (très) difficile...

Tester toutes les colorations possibles et voir si l'une d'entre elles est propre est **impossible** car leur nombre est gigantesque. Pour un graphe à 135 sommets et 4 couleurs, il y a  $4^{135} = 4 \times 4 \times 4 \times \dots \times 4$  possibilités, autant que le nombre d'atomes dans l'univers.

135 fois

## ... et qui ne sera jamais facile ?

Ce problème de coloration est un **problème NP-complet** :

- il existe **un algorithme qui vérifie rapidement** (en temps polynomial) **qu'une solution est valide**,
- on ne connaît **pas d'algorithme qui trouve rapidement** (en temps polynomial) **une solution valide**.

Il est même conjecturé qu'il n'existe pas de tel algorithme.

C'est la fameuse **conjecture P ≠ NP**

C'est un des **sept problèmes du millénaire** avec un million de dollars offert pour sa résolution. On connaît des **milliers de problèmes NP-complets** : coloration de graphe, voyageur de commerce... Prouver l'inexistence (ou l'existence) d'un algorithme rapide **pour n'importe lequel** d'entre eux résoudrait la conjecture et **entraînerait** l'inexistence (ou l'existence) d'un algorithme rapide **pour tous les autres**.

Il est conjecturé que les algorithmes quantiques (utilisés par d'éventuels futurs ordinateurs quantiques) ne permettront pas non plus de résoudre ces problèmes rapidement.

terra-numerica.org

Dans quels cas, ce problème admet-il un algorithme polynomial ?

## Exercice 6 :



# Jeu à 2 joueurs dans les graphes : PRÉVISION

## JEU



Un avion a, par erreur, déversé son chargement de bonbons dans une forêt. Un **enfant** veut sortir de sa maison pour manger un bonbon. Ses **parents** veulent l'en empêcher et doivent ramasser les bonbons avant lui.

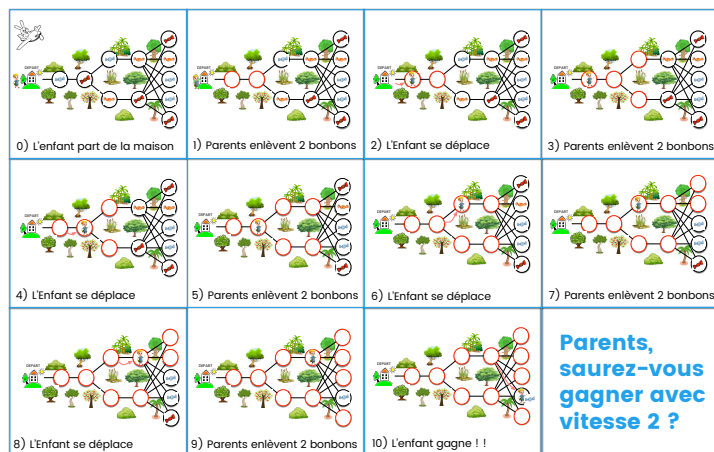


## Règles du jeu

1. Choisir la **vitesse V** des **parents** ( $V = 1$  ou  $2$  ou  $3...$ ).
2. L'**enfant** commence dans la **maison** (DÉPART).
3. Tour à tour :
  - Les **parents** prennent  $V$  bonbons (n'importe où).
  - L'**enfant** se déplace le long d'une arête.

**Fin du jeu** L'**enfant gagne** s'il ramasse un bonbon.  
Sinon, les **parents gagnent**.

## Exemple de déroulement du jeu (avec $V = 2$ )



**Applications :** Un moyen d'accélérer les calculs d'un programme informatique est d'anticiper ses besoins et de « **pré-calculer** » des données. Afin d'**économiser les capacités de calcul**, il n'est pas recommandé de pré-calculer trop de données (on ne peut enlever qu'un nombre limité de bonbons). De plus, les données qui seront effectivement nécessaires ne sont **pas connues à l'avance** (la trajectoire de l'enfant est indéterminée).

**Exemples :** Jeux vidéos, préchargement de pages Web, préchargement de vidéos.