Algorithmique dans l'enseignement secondaire. Quels contenus ? Quel lien aux mathématiques ?

Simon Modeste

Université de Montpellier,
Institut Montpelliérain Alexander Grothendieck (IMAG) U.M.R 5149,
Équipe Didactique et Épistémologie des Mathématiques (DÉMa),

CREM - 6 mars 2020

Contexte français

Algorithmique

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée
- 2012 : Spécialité Informatique et Sciences du Numérique (Term S)

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée
- 2012 : Spécialité Informatique et Sciences du Numérique (Term S)
- 2013 : Rapport de l'académie des sciences sur l'enseignement de l'informatique

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée
- 2012 : Spécialité Informatique et Sciences du Numérique (Term S)
- 2013 : Rapport de l'académie des sciences sur l'enseignement de l'informatique
- 2015 : Informatique et Création Numérique (seconde)

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée
- 2012 : Spécialité Informatique et Sciences du Numérique (Term S)
- 2013 : Rapport de l'académie des sciences sur l'enseignement de l'informatique
- 2015 : Informatique et Création Numérique (seconde)
- 2016 : Informatique au cycle 4 et "Algorithmique et programmation" dans le programme de mathématiques

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée
- 2012 : Spécialité Informatique et Sciences du Numérique (Term S)
- 2013 : Rapport de l'académie des sciences sur l'enseignement de l'informatique
- 2015 : Informatique et Création Numérique (seconde)
- 2016 : Informatique au cycle 4 et "Algorithmique et programmation" dans le programme de mathématiques
- 2017 : Option Informatique au CAPES de Mathématiques

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée
- 2012 : Spécialité Informatique et Sciences du Numérique (Term S)
- 2013 : Rapport de l'académie des sciences sur l'enseignement de l'informatique
- 2015 : Informatique et Création Numérique (seconde)
- 2016 : Informatique au cycle 4 et "Algorithmique et programmation" dans le programme de mathématiques
- 2017 : Option Informatique au CAPES de Mathématiques
- 2019 : Réforme du lycée (évolution de programmes, enseignement de l'informatique)

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée
- 2012 : Spécialité Informatique et Sciences du Numérique (Term S)
- 2013 : Rapport de l'académie des sciences sur l'enseignement de l'informatique
- 2015 : Informatique et Création Numérique (seconde)
- 2016 : Informatique au cycle 4 et "Algorithmique et programmation" dans le programme de mathématiques
- 2017 : Option Informatique au CAPES de Mathématiques
- 2019 : Réforme du lycée (évolution de programmes, enseignement de l'informatique)
- 2020 : CAPES d'informatique (NSI)

Contexte

- 2000 : Rapport Kahane : recommande l'entrée d'une part d'info dans l'enseignement des mathématiques
- 2009/2012 : Algorithmique dans les programmes du lycée
- 2012 : Spécialité Informatique et Sciences du Numérique (Term S)
- 2013 : Rapport de l'académie des sciences sur l'enseignement de l'informatique
- 2015 : Informatique et Création Numérique (seconde)
- 2016 : Informatique au cycle 4 et "Algorithmique et programmation" dans le programme de mathématiques
- 2017 : Option Informatique au CAPES de Mathématiques
- 2019 : Réforme du lycée (évolution de programmes, enseignement de l'informatique)
- 2020 : CAPES d'informatique (NSI)

Collège - Cycle 4

Thème E – Algorithmique et programmation

Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. En créant un programme, ils développent des méthodes de programmation, revisitent les notions de variables et de fonctions sous une forme différente, et s'entraînent au raisonnement.

Attendus de fin de cycle	
• Écrire, mettre au point et exécuter un programme simple	
Connaissances et compétences associées	**Exemples de situations, d'activités et de ressources pour l'élève**
Décomposer un problème en sous-problèmes afin de structurer un programme ; reconnaître des schémas. Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné. Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. Programmer des scripts se déroulant en parallèle. ➢ Notions d'algorithme et de programme. ➢ Notion de variable informatique. ➢ Déclenchement d'une action par un évènement, séquences d'instructions, boucles, instructions conditionnelles. ➢ Notion de message échangé entre objets.	Jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe. Réalisation de figure à l'aide d'un logiciel de programmation pour consolider les notions de longueur et d'angle. Initiation au chiffrement (Morse, chiffre de César, code ASCII…) Construction de tables de conjugaison, de pluriels, jeu du cadavre exquis… Calculs simples de calendrier Calculs de répertoire (recherche, recherche inversée, etc.). Calculs de fréquences d'apparition de chaque lettre dans un texte pour distinguer sa langue d'origine : français, anglais, italien, etc.
Repères de progressivité : En 5ème, les élèves s'initient à la programmation évènementielle. Progressivement, ils développent de nouvelles compétences, en programmant des actions en parallèle, en utilisant la notion de variable informatique, en découvrant les boucles et les instructions conditionnelles qui complètent les structures de contrôle liées aux événements. En 3ème, ils abordent la gestion des objets, en leur faisant échanger des messages.	

Lycée - réforme

Place des mathématiques dans la réforme*

On choisit 3 enseignements de spécialité en 1ère et on en garde 2 en Term. S'ajoutent les option supplémentaires « Mathématiques expertes » et « Mathématiques complémentaires ».

<table>
<tr><td colspan="4">Seconde générale et technologique (mathématiques : 3h/semaine)</td></tr>
<tr><td colspan="3">Première générale
(spécialité mathématiques 4h/semaine)</td><td>Première générale
(sans mathématiques)</td></tr>
<tr><td>Terminale générale
(spé mathématiques : 6h/semaine)</td><td>Terminale générale
(spé mathématiques : 6h/semaine + maths expertes : 3h/semaine)</td><td>Terminale générale
(maths complémentaires : 3h/semaine)</td><td>Terminale générale
(sans mathématiques)</td></tr>
</table>

* il faut lire que d'une année à l'autre on passe d'une case à une case adjacente.

Seconde

4. Algorithmique et programmation

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au cycle 4, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Ce qui est proposé dans ce programme est une consolidation des acquis du cycle 4 autour de deux idées essentielles :

- la notion de fonction d'une part, et
- la programmation comme production d'un texte dans un langage informatique d'autre part.

Dans le cadre de cette activité, les élèves sont entraînés :

- à décrire des algorithmes en langage naturel ou dans un langage de programmation ;
- à en réaliser quelques-uns à l'aide d'un programme simple écrit dans un langage de programmation textuel;
- à interpréter des algorithmes plus complexes.

Un langage de programmation simple d'usage est nécessaire pour l'écriture des programmes. Le choix du langage se fera parmi les langages interprétés, concis, largement répandus, et pouvant fonctionner dans une diversité d'environnements.

L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes ainsi traités doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante.

À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de donner aux élèves de bonnes habitudes de rigueur et de les entraîner aux pratiques systématiques de vérification et de contrôle. En programmant, les élèves revisitent les notions de variables et de fonctions sous une forme différente. Il convient d'y être attentif.

Seconde

CONTENUS	CAPACITÉS ATTENDUES	COMMENTAIRES
Variables et instructions élémentaires	• choisir ou déterminer le type d'une variable (entier, flottant ou chaîne de caractères) ; • concevoir et écrire des affectations à des variables ; • écrire une formule permettant un calcul combinant des variables.	On commence par consolider les notions de variables, de boucles et d'instructions conditionnelles introduites au cycle 4 en complétant la programmation par blocs par l'utilisation d'un langage de programmation textuel.
Boucle et itérateur, instruction conditionnelle	• programmer une instruction conditionnelle ; • programmer une boucle bornée ; • programmer une boucle non bornée.	On formalise les notions de boucle bornée (for) et de boucle non bornée (while) et on introduit la notion nouvelle de fonction dans un langage de programmation.
Notion de fonction	• programmer des fonctions simples, ayant un petit nombre d'arguments.	Il est intéressant de confronter les fonctions dans un langage de programmation avec les fonctions d'un tableur.

Première

Algorithmique et programmation

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. La classe de seconde a permis de consolider les acquis du cycle 4 autour de deux idées essentielles :

- la notion de fonction ;
- la programmation comme production d'un texte dans un langage informatique.

L'enseignement de spécialité de mathématiques de classe de première vise la consolidation des notions de variable, d'instruction conditionnelle et de boucle ainsi que l'utilisation des fonctions. La seule notion nouvelle est celle de liste qui trouve naturellement sa place dans de nombreuses parties du programme et aide à la compréhension de notions mathématiques telles que les suites numériques, les tableaux de valeurs, les séries statistiques…

Comme en classe de seconde, les algorithmes peuvent être écrits en langage naturel ou utiliser le langage Python.

Les notions relatives aux types de variables et à l'affectation sont consolidées. Comme en classe de seconde, on utilise le symbole « $\leftarrow$ » pour désigner l'affection dans un algorithme écrit en langage naturel.

L'accent est mis sur la programmation modulaire qui permet de découper une tâche complexe en tâches plus simples.

Première

- **Histoire des mathématiques**

De nombreux textes témoignent d'une préoccupation algorithmique au long de l'Histoire. Lorsqu'un texte historique a une visée algorithmique, transformer les méthodes qu'il présente en un algorithme, voire en un programme, ou inversement, est l'occasion de travailler des changements de registre qui donnent du sens au formalisme mathématique.

- **Notion de liste**

La génération des listes en compréhension et en extension est mise en lien avec la notion d'ensemble. Les conditions apparaissant dans les listes définies en compréhension permettent de travailler la logique. Afin d'éviter des confusions, on se limite aux listes sans présenter d'autres types de collections.

Capacités attendues

- Générer une liste (en extension, par ajouts successifs ou en compréhension).
- Manipuler des éléments d'une liste (ajouter, supprimer…) et leurs indices.
- Parcourir une liste.
- Itérer sur les éléments d'une liste.

Terminale

Algorithmique et programmation

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Les classes de seconde et de première ont permis de consolider les acquis du collège (notion de variable, type, de variables, affectation, instruction conditionnelle, boucle notamment), d'introduire et d'utiliser la notion de fonction informatique et de liste. En algorithmique et programmation, le programme reprend les programmes de seconde et de première sans introduire de notion nouvelle, afin de consolider le travail des classes précédentes.

Les algorithmes peuvent être écrits en langage naturel ou utiliser le langage Python. On utilise le symbole « $\leftarrow$ » pour désigner l'affection dans un algorithme écrit en langage naturel. L'accent est mis sur la programmation modulaire qui permet de découper une tâche complexe en tâches plus simples.

L'algorithmique trouve naturellement sa place dans toutes les parties du programme et aide à la compréhension et à la construction des notions mathématiques.

Terminale

- **Histoire des mathématiques**

De nombreux textes témoignent d'une préoccupation algorithmique au long de l'Histoire. Lorsqu'un texte historique a une visée algorithmique, transformer les méthodes qu'il présente en un algorithme, voire en un programme, ou inversement, est l'occasion de travailler des changements de registre qui donnent du sens au formalisme mathématique.

Notion de liste

La génération des listes en compréhension et en extension est mise en lien avec la notion d'ensemble. Les conditions apparaissant dans les listes définies en compréhension permettent de travailler la logique. Afin d'éviter des confusions, on se limite aux listes sans présenter d'autres types de collections.

Capacités attendues

- Générer une liste (en extension, par ajouts successifs ou en compréhension).
- Manipuler des éléments d'une liste (ajouter, supprimer…) et leurs indices.
- Parcourir une liste.
- Itérer sur les éléments d'une liste.

Questions

- Quelles interactions entre les sciences informatique et mathématiques ?

Questions

- Quelles interactions entre les sciences informatique et mathématiques ?
- Quelles interactions/synergies dans l'enseignement ?

Questions

- Quelles interactions entre les sciences informatique et mathématiques ?
- Quelles interactions/synergies dans l'enseignement ?

$\rightarrow$ Plusieurs angles d'attaque : logique et fondements, mathématiques de l'informatique/mathématiques discrètes, apports de l'informatique aux mathématiques et évolutions des pratiques en mathématiques, ...

Questions

- Quelles interactions entre les sciences informatique et mathématiques ?
- Quelles interactions/synergies dans l'enseignement ?

$\rightarrow$ Plusieurs angles d'attaque : logique et fondements, mathématiques de l'informatique/mathématiques discrètes, apports de l'informatique aux mathématiques et évolutions des pratiques en mathématiques, ...
$\rightarrow$ algorithmique (travail de thèse), ...

Qu’est-ce que l’algorithmique ?

Qu'est-ce que l'algorithmique ?

Un détour par l'épistémologie....

Qu'est-ce que l'algorithmique ?

Un détour par l'épistémologie....

"Comme connaissance des processus par lesquels les concepts mathématiques se forment et se développent et plus généralement comme connaissance des caractéristiques de l'activité mathématique"

Au delà de l'analyse conceptuelle, l'épistémologie intervient à ce niveau sur un plan plus général car ce que vise l'enseignement des mathématiques, ce n'est pas simplement la transmission de connaissances mathématiques, c'est plus globalement celle d'une culture. Il s'agit de faire entrer les élèves dans le jeu mathématique. Mais, qu'est ce jeu mathématique ? Quels sont les processus généraux de pensée qui le gouvernent ? C'est l'analyse épistémologique [...] qui est au premier chef concernée par ces questions.
(Artigue, 1990, p. 246)

L'analyse épistémologique permet également au didacticien de prendre la mesure des disparités existant entre savoir "savant" [...] et savoir "enseigné". En effet, alors que l'école vit sur la fiction consistant à voir dans les objets d'enseignement des copies simplifiées mais fidèles des objets de la science, l'analyse épistémologique, en nous permettant de comprendre ce qui gouverne l'évolution de la connaissance scientifique, nous aide à prendre conscience de la distance qui sépare les économies des deux systèmes.
(Artigue, 1990, p. 244-245)

Exemples

Algorithme

Définition retenue :

Un **algorithme** est une procédure de résolution de problème, s'appliquant à une famille d'instances du problème et produisant, en un nombre fini d'étapes constructives, non-ambigües et organisées, la réponse au problème pour toute instance de cette famille.

Exemples : division euclidienne, méthode du pivot de Gauss, test de primalité, tris, recherche du couple de points les plus proches , parcours de graphes, outils de calculs formels, etc.

Recherche d'une frontière

Recherche d'une frontière

Recherche d'une frontière

Recherche d'une frontière

Recherche d'une frontière

- On peut interroger une case à la fois

Recherche d'une frontière

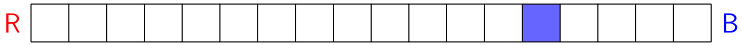
- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

R														?					B

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière



- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?
- … en interrogeant le moins de cases possible ?

Recherche d'une frontière

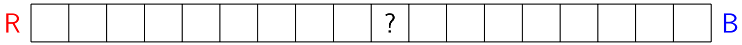
- On peut interroger une case à la fois
- Comment retrouver la frontière ?
- ... en interrogeant le moins de cases possible ?
- Algorithme de recherche par dichotomie :

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?
- ... en interrogeant le moins de cases possible ?
- Algorithme de recherche par dichotomie :

  On teste la case qui sépare le territoire de la manière la plus équilibrée.
  On réitère sur le territoire restant jusqu'à trouver la frontière.

Recherche d'une frontière



- On peut interroger une case à la fois
- Comment retrouver la frontière ?
- … en interrogeant le moins de cases possible ?
- Algorithme de recherche par dichotomie :

  On teste la case qui sépare le territoire de la manière la plus équilibrée.
  On réitère sur le territoire restant jusqu'à trouver la frontière.

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?
- … en interrogeant le moins de cases possible ?
- Algorithme de recherche par dichotomie :

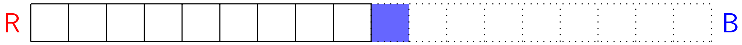
  On teste la case qui sépare le territoire de la manière la plus équilibrée.
  On réitère sur le territoire restant jusqu'à trouver la frontière.

Recherche d'une frontière

- On peut interroger une case à la fois
- Comment retrouver la frontière ?
- ... en interrogeant le moins de cases possible ?
- Algorithme de recherche par dichotomie :

  On teste la case qui sépare le territoire de la manière la plus équilibrée.
  On réitère sur le territoire restant jusqu'à trouver la frontière.

Recherche d'une frontière



- On peut interroger une case à la fois
- Comment retrouver la frontière ?
- ... en interrogeant le moins de cases possible ?
- Algorithme de recherche par dichotomie :

  On teste la case qui sépare le territoire de la manière la plus équilibrée.
  On réitère sur le territoire restant jusqu'à trouver la frontière.

Variante

- On interroge une case à la fois

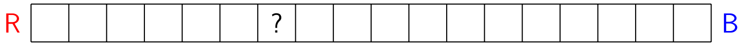
Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.

Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante

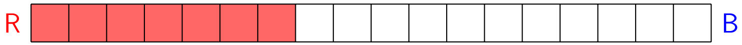


- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante



- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

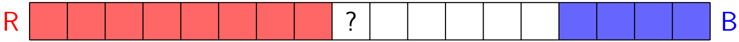
Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante

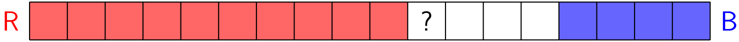


- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante

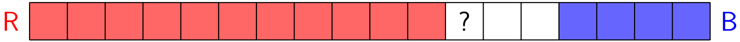
- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante



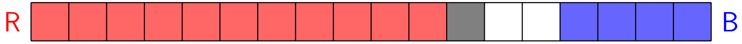
- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante



- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante



- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?

Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?
- ... en interrogeant le moins de cases possible ?

Variante

- On interroge une case à la fois
- On n'a pas le droit de tomber plus d'une fois sur une case bleue.
- Comment retrouver la frontière ?
- ... en interrogeant le moins de cases possible ?

  Partager en deux territoires équilibrés ne produit plus l'algorithme optimal.

Existence du PGCD (1)

Preuve :
Soient a et b deux entiers avec $a \geqslant b$. Notons D l'ensemble des diviseurs communs à a et b. Comme $1|a$ et $1|b$, on a $D \neq \emptyset$, et $D \subset \{1, ..., b\}$. D est borné non vide, donc possède un plus grand élément d noté $pgcd(a, b)$.

Existence du PGCD (2)

Preuve : Démontrons la propriété "$\forall (m,n) \in \mathbb{N}^2$, $pgcd(m,n)$ existe", par récurrence sur $\min(m,n)$.

• Initialisation : Soient $(a,b) \in \mathbb{N}^2$ tel que $\min(a,b) = 0$, i.e. $a = 0$ ou $b = 0$. Alors $pgcd(a,b)$ existe et vaut $\max(a,b)$.

• Hérédité : Supposons que $\forall (m,n)$ tel que $\min(m,n) \leqslant p$, $pgcd(m,n)$ existe.
Soit un couple (a,b) tel que $\min(a,b) = p+1$. Supposons, par exemple, $b \leqslant a$ et posons r le reste de la division de a par $b = p+1$. Alors $pgcd(a,b) = pgcd(b,r)$ et comme $r < p+1$, par hypothèse de récurrence, $pgcd(a,b)$ existe.

Algorithme du PGCD

Données : a, b, deux entiers

$r_1 \leftarrow a$;

$r_2 \leftarrow b$;

tant que $r_2 \neq 0$ **faire**

  la division euclidienne donne $r_1 = q.r_2 + r$;

  $r_1 \leftarrow r_2$;

  $r_2 \leftarrow r$;

Résultat : r_1

Algorithme 1 : Euclide

Algorithme pour l'enveloppe convexe

- Problème :

Algorithme pour l'enveloppe convexe

- Problème :

Entrée : un ensemble fini de points du plans

Algorithme pour l'enveloppe convexe

- Problème :

Entrée : un ensemble fini de points du plans

Sortie : une liste ordonnée des points de l'enveloppe convexe

Algorithme pour l'enveloppe convexe

- Problème :

Entrée : un ensemble fini de points du plans
Sortie : une liste ordonnée des points de l'enveloppe convexe

- Plusieurs algorithmes existent : diviser pour régner, Jarvis ("paquet cadeau"), Graham...

Algorithme pour l'enveloppe convexe

- Problème :

Entrée : un ensemble fini de points du plans
Sortie : une liste ordonnée des points de l'enveloppe convexe

- Plusieurs algorithmes existent : diviser pour régner, Jarvis ("paquet cadeau"), Graham...
- Complexité des algorithmes – complexité du problème

Dixième problème de Hilbert

- Problème (1900) :

Dixième problème de Hilbert

- Problème (1900) :

Entrée : Un entier n et un polynôme P à coefficients dans $\mathbb{Z}$, à n inconnues

Dixième problème de Hilbert

- Problème (1900) :

Entrée : Un entier n et un polynôme P à coefficients dans $\mathbb{Z}$, à n inconnues

Question : L'équation en $X_1, ..., X_n$ $P(X_1, ..., X_n)$ possède-t-elle une solution dans $\mathbb{Z}^n$?

Dixième problème de Hilbert

- Problème (1900) :

Entrée : Un entier n et un polynôme P à coefficients dans $\mathbb{Z}$, à n inconnues

Question : L'équation en $X_1, ..., X_n$ $P(X_1, ..., X_n)$ possède-t-elle une solution dans $\mathbb{Z}^n$?

- On ne peut pas trouver d'algorithme permettant de résoudre ce problème (Matiiassevitch, 1970)

Dixième problème de Hilbert

- Problème (1900) :

Entrée : Un entier n et un polynôme P à coefficients dans $\mathbb{Z}$, à n inconnues

Question : L'équation en $X_1, ..., X_n$ $P(X_1, ..., X_n)$ possède-t-elle une solution dans $\mathbb{Z}^n$?

- On ne peut pas trouver d'algorithme permettant de résoudre ce problème (Matiiassevitch, 1970)
- Modèles théoriques de ce qui est algorithmique

Problème = Instances + Question

Problème = Instances + Question

Problème (dans la théorie de la complexité algorithmique) :

Problème

I : un ensemble d'instances
Q : une question

Problème = Instances + Question

Problème (dans la théorie de la complexité algorithmique) :

P : Calcul du pgcd

I : l'ensemble des couples (a, b) d'entiers naturels (i.e. $\mathbb{N}^2$)
Q : quel est le pgcd de a et b ?

Problème = Instances + Question

Problème (dans la théorie de la complexité algorithmique) :

P : Résolution de systèmes d'équations linéaires

I : les systèmes d'équations linéaires
Q : quel est l'ensemble des solutions du système ?

Problème = Instances + Question

Problème (dans la théorie de la complexité algorithmique) :

Problème

I : un ensemble d'instances
Q : une question

- **Résoudre le problème algorithmiquement**, c'est fournir un algorithme qui donne la réponse à la question Q quelle que soit l'instance proposée

Problème = Instances + Question

Problème (dans la théorie de la complexité algorithmique) :

Problème

I : un ensemble d'instances
Q : une question

- **Résoudre le problème algorithmiquement**, c'est fournir un algorithme qui donne la réponse à la question Q quelle que soit l'instance proposée
- **Prouver un algorithme**, c'est prouver qu'il donne toujours (i.e. après un nombre fini d'étapes) une réponse correcte pour toute instance du problème

Généralité

Généralité

Un algorithme ne résout donc pas un problème unique mais toute une classe de problèmes ne différent que par les donnés mais gouvernés par les mêmes prescriptions. (Bouvier et al., 2005, p 27)

Expression

- Opérateurs, exécuteur, technologie : quelque soit l'opérateur, s'il peut effectuer les opérations élémentaires de l'algorithme, les étapes et le résultat sera toujours le même.
- Opérations simples/élémentaires
- Non-ambiguës
- Description finie
- Exécutions finies (mais peuvent être aussi grandes que l'on veut...)

Algorithme et preuve

Algorithme et preuve

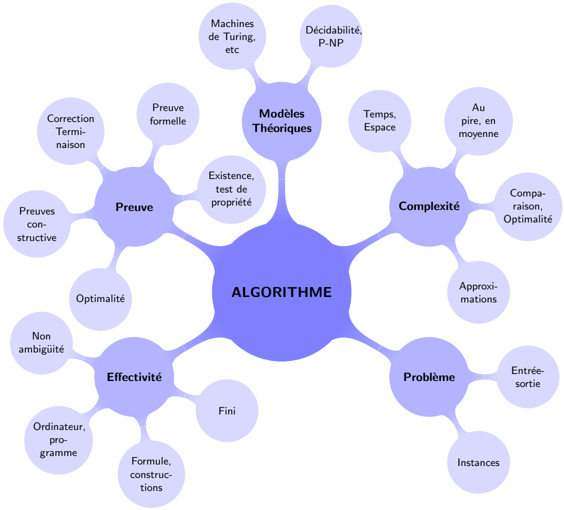
- Preuve d'un algorithme : Correction et terminaison
- Preuves algorithmiques : récurrences, constructions explicites, etc.
  Existence d'objet ou reconnaissance de propriété.
- Preuve des propriétés d'algorithmes
- Algorithme comme outils de preuve, étape d'une preuve.
- Algorithmes de vérification de preuve et de recherche de preuve

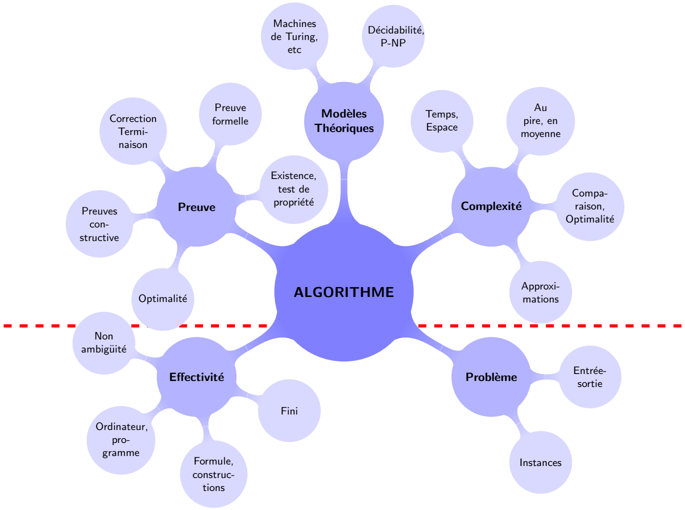
Complexité

- Un élément spécifique de la pensée algorithmique (Knuth)
- Complexité en temps : nombre d'étapes nécessaire en fonction de la taille de l'instance.
- Complexité en espace : quantité d'espace nécessaire en fonction de la taille de l'instance.
- Complexité évaluée "au pire" ou "en moyenne"
- Comparaison d'algorithmes, optimalité d'un algorithme
- Complexité d'un problème
- Classes de complexité, familles de problèmes

Développements théoriques

- Modèles de ce qui est algorithmiquement calculable (machine de Turing, fonctions récursives,...)
- Théories de la calculabilité et de la complexité
- Champ mathématiques/informatique théorique à part entière


ALGORITHME
Modèles Théoriques
Machines de Turing, etc
Décidabilité, P-NP
Complexité
Temps, Espace
Au pire, en moyenne
Compa-raison, Optimalité
Approxi-mations
Problème
Entrée-sortie
Instances
Effectivité
Non ambigüité
Ordinateur, pro-gramme
Formule, construc-tions
Fini
Preuve
Correction Termi-naison
Preuve formelle
Existence, test de propriété
Preuves con-structive
Optimalité



- ☺ Dualité outil-objet

3 paradigmes

3 formes pour l'algorithme :

- Preuve Algorithmique

  Algorithme et preuve mêlés (récurrence, ...), langage mathématiques

- Algorithme Mathématique

  Preuve et algorithme dissociés, langage mathématiques et éléments spécifiques issu de la programmation

- Algorithme Informatique

  Algorithme en langage de programmation, dissocié de sémantique et validation

Questions didactiques

Transposition didactique

Phénomènes et effets du passage du savoir savant au savoir à enseigner puis au savoir enseigné.

Transposition didactique

Phénomènes et effets du passage du savoir savant au savoir à enseigner puis au savoir enseigné.

Particulièrement visible si l'on compare la transposition didactique de l'algorithme entre programmes de mathématiques du lycée et de la spécialité ISN.

Deux transpositions distinctes :

Deux transpositions distinctes :

- Programmes de mathématiques du lycée :

Deux transpositions distinctes :

- Programmes de mathématiques du lycée :
    - Algorithme non-défini
    - Algorithmique comme “démarche”
    - Algorithme peu distingué de programme
    - Algorithme uniquement outil
    - Activité sur “langage” (pour programmer ensuite)
    - Paradigme AI

Deux transpositions distinctes :

- Programmes de mathématiques du lycée :
  - Algorithme non-défini
  - Algorithmique comme "démarche"
  - Algorithme peu distingué de programme
  - Algorithme uniquement outil
  - Activité sur "langage" (pour programmer ensuite)
  - Paradigme AI
- Programmes de spécialité ISN :

Deux transpositions distinctes :

- Programmes de mathématiques du lycée :
  - Algorithme non-défini
  - Algorithmique comme “démarche”
  - Algorithme peu distingué de programme
  - Algorithme uniquement outil
  - Activité sur “langage” (pour programmer ensuite)
  - Paradigme AI
- Programmes de spécialité ISN :
  - Définition de l’algorithme
  - Algorithmique comme branche
  - Distinction nette algorithme/programme
  - Algorithme outil et objet (complexité)
  - Algorithmes non limités à la programmation
  - Paradigmes AI et AM

En mathématiques :

Algorithmique (objectifs pour le lycée)

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, les élèves ont rencontré des algorithmes (algorithmes opératoires, algorithme des différences, algorithme d'Euclide, algorithmes de construction en géométrie). Ce qui est proposé dans le programme est une formalisation en langage naturel propre à donner lieu à traduction sur une calculatrice ou à l'aide d'un logiciel. Il s'agit de familiariser les élèves avec les grands principes d'organisation d'un algorithme : gestion des entrées-sorties, affectation d'une valeur et mise en forme d'un calcul.

Dans le cadre de cette activité algorithmique, les élèves sont entraînés :

- à décrire certains algorithmes en langage naturel ou dans un langage symbolique ;
- à en réaliser quelques uns à l'aide d'un tableur ou d'un petit programme réalisé sur une calculatrice ou avec un logiciel adapté ;
- à interpréter des algorithmes plus complexes.

Aucun langage, aucun logiciel n'est imposé.

L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes posés doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante.

À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de donner aux élèves de bonnes habitudes de rigueur et de les entraîner aux pratiques systématiques de vérification et de contrôle.

En mathématiques :

La grille ci-dessous peut donner quelques éléments en ce sens, non limitatifs il va de soi, et on adaptera ce questionnement aux situations effectivement rencontrées.

Critère	Excellent	Bon	Moyen	Insuffisant
Respect des bons usages Le but visé par l'algorithme est explicité, et des commentaires précisent le déroulement. Les variables ont des noms bien choisis.	Aucune erreur	De petits détails sont négligés. Le but est difficile à déterminer	Des détails manquent, mais le programme tente quand même d'accomplir ses fonctions essentielles.	Ne répond pas au problème posé. Objectif impossible à déterminer
Correction du code : L'algorithme fonctionne.	Fonctionne correctement dans **tous** les cas.	Fonctionne pour des données (entrées) standard mais échecs mineurs sur des cas particuliers.	Échoue pour des données (entrées) standard, mais pour une raison mineure.	Échoue pour des données (entrées) standard, pour une raison importante.
Interface utilisateur : (entrées, sorties) Elle est claire et commode.	Aucune faute	1-3 fautes mineures	Plus de trois fautes mineures ou une faute majeure	Plus d'une faute majeure

En spécialité ISN :

4.2 Algorithmique

Un algorithme se définit comme une méthode opérationnelle permettant de résoudre, en un nombre fini d'étapes clairement spécifiées, toutes les instances d'un problème donné. Cette méthode peut être exécutée par une machine ou par une personne.

Les élèves ont été confrontés aux algorithmes très tôt dans leur parcours scolaire (avec les quatre opérations arithmétiques) et régulièrement de nouvelles situations de nature algorithmique leur ont été proposées ; ainsi, la construction de figures en géométrie euclidienne, la transcription des « formules » moléculaires en chimie, le code génétique ou encore l'analyse fonctionnelle en technologie sont autant de situations évoquant des algorithmes. Les

On présente simultanément les notions d'algorithme et de programme, puis on les distingue.
L'objectif est une compréhension de ces algorithmes et la capacité à les mettre en œuvre.

Un exemple au lycée : l'algorithme instancié.

Un produit de la transposition didactique :

Un produit de la transposition didactique :

- **Algorithme-instancié :**
  Expression d'un algorithme dont les variables représentant l'entrée ont été remplacées par des valeurs précises.
  Un algorithme-instancié est à mi-chemin entre un algorithme et son exécution sur une instance.

```
Mettre 5000 dans S
Mettre 0 dans N
Effacer l'écran
Tant que S est strictement inférieur à 8000
        Remplacer S par S*1,02
        Augmenter N de 1
        Afficher N et S
Fin du Tant Que
```

Mettre 5000 dans S
Mettre 0 dans N
Effacer l'écran
Tant que S est strictement inférieur à 8000
    Remplacer S par S*1,02
    Augmenter N de 1
    Afficher N et S
Fin du Tant Que

```
début
    Donner à res la valeur 1
    pour i de 1 à 10 faire
        Donner à res la valeur res * i
    fin
    Afficher res
fin
```

Algorithme 7 : Factorielle "Pour"

```
début
  Donner à a la valeur 0    Donner à b la valeur 10
  Donner à N la valeur 50    Donner à pas la valeur (b − a)/N
  Donner à x la valeur a    Donner à max la valeur 2x^2 − 5x + 3
  pour i de 1 à N faire
    Donner à x la valeur x + pas
    Donner à y la valeur 2x^2 − 5x + 3
    si max < y alors
      Donner à max la valeur y
    fin
  fin
  Afficher max
fin
```

Algorithme 13 : Inconnu

Conditions et contraintes

Conditions et contraintes

- Conditions : Existe-t-il un milieu adapté à la vie de l'algorithme-objet dans les programmes du lycée ?

Conditions et contraintes

- Conditions : Existe-t-il un milieu adapté à la vie de l'algorithme-objet dans les programmes du lycée ?
- Contraintes : Quelles contraintes pèsent sur le système didactique et influencent la transposition didactique au lycée ?

Conditions et contraintes

- Conditions : Existe-t-il un milieu adapté à la vie de l'algorithme-objet dans les programmes du lycée ?
- Contraintes : Quelles contraintes pèsent sur le système didactique et influencent la transposition didactique au lycée ?

Conditions et contraintes

- Conditions : Existe-t-il un milieu adapté à la vie de l'algorithme-objet dans les programmes du lycée ?
- Contraintes : Quelles contraintes pèsent sur le système didactique et influencent la transposition didactique au lycée ?

L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes posés doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante. [Programmes du lycée]

Les suites en 1ère ES

Les suites en 1ère ES

54 ALGO

Soit (u_n) la suite définie sur $\mathbb{N}$ par son premier terme u_0 et la relation $u_{n+1} = 2u_n + 5$.

1. Écrire un programme permettant de calculer le terme d'indice N donné de cette suite.

2. Déterminer le terme d'indice 11 de (u_n) lorsque $u_0 = 1$.

→ Pour vous aider Savoir-faire 3, p. 137

55 ALGO

Soit (u_n) la suite définie sur $\mathbb{N}$ par son premier terme u_0 et la relation $u_{n+1} = \dfrac{u_n + 3}{3u_n^2 + 1}$.

1. Écrire un programme permettant de calculer le terme d'indice N donné de cette suite.

2. Déterminer le terme d'indice 17 de (u_n) lorsque $u_0 = -1$.

→ Pour vous aider Savoir-faire 3, p. 137

Les suites en 1ère ES

13 ALGO

L'algorithme suivant définit une suite.

```
Saisir A
Saisir N
  Pour I variant de 1 à N
  |     A prend la valeur 5 × A − 3
  Fin Pour
Afficher A
```

1. Si l'on entre $A = 1$ et $N = 2$, quelle valeur de A sera affichée après l'exécution de l'algorithme ?

2. Quelle valeur de N faut-il saisir pour obtenir le troisième terme ?

14 ALGO

On considère l'algorithme suivant :

```
Saisir A
Saisir N
  Pour I variant de 1 à N
  |     A prend la valeur 2 × A − 1
  Fin Pour
Afficher A
```

Quelle valeur de A sera affichée après exécution de l'algorithme :

a. si on entre $A = 1$ et $N = 5$?

b. si on entre $A = 2$ et $N = 3$?

Les suites en 1ère ES

50 ALGO

La suite u_n est définie par $u_0 = A$ et l'algorithme suivant permettant d'afficher les termes de u_1 à u_N.

```
Saisir A
Saisir N
  Pour I variant de 1 à N
  |    A prend la valeur 2 × A − 1
  Fin Pour
Afficher A
```

1. Déterminer u_1, u_2, u_3 et u_4 quand $u_0 = 3$.
2. Exprimer u_{n+1} en fonction de u_n.
3. Représenter les cinq premiers termes de la suite dans un repère (O ; I, J).

Les suites en 1ère ES

100 ALGO

Soit u la suite définie pour tout entier naturel n par $u_n = \frac{1}{3^n}$.

1. Déterminer le sens de variation de u.

2. Déterminer un entier n_0 tel que pour tout $n \geqslant n_0$:

$$0 < u_n \leqslant 10^{-3}.$$

3. Écrire un algorithme donnant le plus petit entier n tel que :

$$u_n \leqslant 10^{-90}.$$

AIDE : On pourra utiliser la boucle « **tant que** ».

101 ALGO

1. Faire fonctionner « à la main » l'algorithme ci-dessous.

Initialisation
N prend la valeur 0
U prend la valeur 10
Traitement
  Tant que $U \leqslant 100$
    N prend la valeur $N + 1$
    U prend la valeur $2U - 5$
  Fin Tant que
Sortie
Afficher N

Les suites en 1ère ES

Exemple d'algorithme permettant d'obtenir des termes d'une suite

Soit (u_n) la suite définie par : $u_0 = A$ et $u_{n+1} = 3u_n - 1$ pour $n \in \mathbb{N}$.

L'algorithme suivant permet d'obtenir le terme d'un rang donné de la suite (u_n).

```
Saisir A
Saisir N
  Pour I variant de 1 à N
  |     A prend la valeur 3 × A – 1
  Fin Pour
Afficher A
```

La valeur de u_0 est entrée dans A.

Dans la boucle « Pour », on calcule d'abord $3u_0 - 1$, c'est-à-dire u_1.

Puis de la même façon, on calcule u_2, u_3…

L'indice du terme de la suite peut être affiché en ajoutant l'instruction : « Afficher I » en dehors de la boucle avant « Afficher A ».

Bac ES - 2014

À l'automne 2010, Claude achète une maison à la campagne ; il dispose d'un terrain de 1500m^2 entièrement engazonné. Mais tous les ans, 20% de la surface engazonnée est détruite et remplacée par de la mousse. Claude arrache alors, à chaque automne, la mousse sur une surface de 50m^2 et la remplace par du gazon.

Pour tout nombre entier naturel n, on note u_n la surface en m^2 de terrain engazonné au bout de n années, c'est-à-dire à l'automne 2010 + n. On a donc $u_0 = 1\,500$.

1. Calculer u_1.

2. Justifier que, pour tout nombre entier naturel n, $u_{n+1} = 0{,}8\,u_n + 50$.

3. On considère la suite (v_n) définie pour tout nombre entier naturel n par : $v_n = u_n - 250$.

a) Démontrer que la suite (v_n) est géométrique. Préciser son premier terme et sa raison.

b) Exprimer v_n en fonction de n.

En déduire que, pour tout nombre entier naturel n, $u_n = 250 + 1\,250 \times 0{,}8^n$.

c) Quelle est la surface de terrain engazonné au bout de 4 années ?

4. ***a)*** Déterminer par le calcul la plus petite valeur de l'entier naturel n telle que :

$$250 + 1\,250 \times 0{,}8^n < 500$$

Interpréter le résultat obtenu.

b) Compléter l'algorithme fourni en **annexe 1** pour qu'il affiche la solution obtenue à la question précédente.

Thème E – Algorithmique et programmation

Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. En créant un programme, ils développent des méthodes de programmation, revisitent les notions de variables et de fonctions sous une forme différente, et s'entraînent au raisonnement.

<table>
<tr><th colspan="2">Attendus de fin de cycle</th></tr>
<tr><td colspan="2">• Écrire, mettre au point et exécuter un programme simple</td></tr>
<tr><th>Connaissances et compétences associées</th><th>Exemples de situations, d'activités et de ressources pour l'élève</th></tr>
<tr><td>Décomposer un problème en sous-problèmes afin de structurer un programme ; reconnaître des schémas.
Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné.
Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.
Programmer des scripts se déroulant en parallèle.
➢ Notions d'algorithme et de programme.
➢ Notion de variable informatique.
➢ Déclenchement d'une action par un évènement, séquences d'instructions, boucles, instructions conditionnelles.
➢ Notion de message échangé entre objets.</td><td>Jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe.
Réalisation de figure à l'aide d'un logiciel de programmation pour consolider les notions de longueur et d'angle.
Initiation au chiffrement (Morse, chiffre de César, code ASCII…)
Construction de tables de conjugaison, de pluriels, jeu du cadavre exquis…
Calculs simples de calendrier
Calculs de répertoire (recherche, recherche inversée, etc.).
Calculs de fréquences d'apparition de chaque lettre dans un texte pour distinguer sa langue d'origine : français, anglais, italien, etc.</td></tr>
<tr><td colspan="2">Repères de progressivité :
En $5^{\text{ème}}$, les élèves s'initient à la programmation évènementielle. Progressivement, ils développent de nouvelles compétences, en programmant des actions en parallèle, en utilisant la notion de variable informatique, en découvrant les boucles et les instructions conditionnelles qui complètent les structures de contrôle liées aux événements. En $3^{\text{ème}}$, ils abordent la gestion des objets, en leur faisant échanger des messages.</td></tr>
</table>

Et au collège ?

Et au collège ?

- Quelle continuité collège-lycée ?
- Quelle articulation entre numérique/algorithmique/algébrique ?
- Quelle articulation algorithmique/programmation ?
- Quelle articulation avec les autres thèmes mathématiques ?
- Quelle formation des enseignants ?

Merci.