
VIE DANS LES IREM

*Introduction de
l'algorithmique en Seconde :
Réflexions et propositions
du groupe AMECMI de l'Irem de Lille*

Le groupe AMECMI [Activités Mathématiques pour Enseigner en Classe avec un Média Informatique] de L'Irem de Lille a réagi sans tarder à l'annonce de l'introduction de l'algorithmique en Seconde : alors même que le programme était encore en discussion, il s'est réuni pour défricher le terrain et produire un premier texte qui rendra service à tous les enseignants appelés à introduire leurs élèves de Seconde dans cette nouvelle démarche. L'article qui suit est modeste aux dires même des auteurs, il cherche à donner aux collègues quelques clés et à rassurer ceux qui craignent ce nouvel enseignement : le lecteur y trouve « des remarques simples sur des choses qui sont essentiellement simples, au niveau de la Seconde. » Ce texte est une première étape : les membres du groupe appellent ceux qui le souhaitent à les rejoindre pour travailler avec eux en présentiel ou à distance sur le thème amorcé. Une démarche exemplaire.

Le comité de rédaction de Repères-Irem

Introduction

Le nouveau programme de mathématiques de Seconde applicable à la rentrée 2009 est paru à l'adresse :

<http://media.education.gouv.fr/file...>

Il comprend une section d'algorithmique. Le groupe AMECMI de l'Irem de Lille, renforcé de collègues invités, que nous remercions ici, a tenu une réunion à ce sujet le 15 mai dernier et a cru utile de rédiger, à l'intention des professeurs débutants dans ce domaine (ou qui se considéreraient comme tels), le texte qui suit. Il ne s'agit pas d'un texte savant ou technique, mais de remarques simples sur des choses qui sont essentiellement simples, au niveau de la Seconde.

Nous conseillons même de commencer l'enseignement d'algorithmique de façon élémentaire, voir : *Un exemple introductif à l'algorithmique sur machine*, [EO], développé par Emmanuel Ostenne, qui traite un même algorithme très simple avec successivement la quasi-totalité des matériels et logiciels existants, afin de permettre aux professeurs de se faire une idée sur plusieurs langages candidats à l'enseignement de l'algorithmique en Seconde, à l'adresse :

<http://emmanuel.ostenne.free.fr/mep...>

Ceci dit, les professeurs devraient savoir comment l'algorithmique se développe dans les cursus concernés, où elle mène, quel type de problème elle permet de traiter, quelle est sa place dans les mathématiques (et l'informatique). Cela pose la question bibliographique.

En annexe de ce texte auquel elle apporte de la profondeur, Alain Juhel a préparé une

« Bibliographie amoureuse de l'algorithmique » [EJ] disponible à l'adresse :

http://home.nordnet.fr/ajuhel/Z_Al...

Cette bibliographie commentée en abordant des sujets comme les structures de données et tous les niveaux d'enseignement devrait permettre au professeur débutant comme au pratiquant confirmé de trouver l'ouvrage qui lui conviendra.

I. — Qu'est-ce que l'algorithmique ?

1. Les algorithmes dans la vie courante

Les manuels d'utilisation des machines électroniques actuelles sont essentiellement des recueils d'algorithmes. On pourrait multiplier les exemples. Examinons plus particulièrement des algorithmes plus anciens : les recettes de cuisine.

Une recette de cuisine comporte 3 parties :

1. Réunir les ingrédients
2. Préparer
3. Déguster

La préparation consiste à exécuter une suite d'instructions : par exemple, plonger les tomates dans une casserole d'eau bouillante pendant quelques instants avant de les peler. On ne sait pas pourquoi il faut procéder de la sorte et d'ailleurs, ça n'a aucune importance. La recette a été écrite par quelqu'un qui sait. Elle marche.

En pensant aux algorithmes de mathématiques, on pourrait dire que les ingrédients de la recette sont les entrées du processus auxquelles on applique l'algorithme (la préparation) pour obtenir, en sortie, un plat que l'on dégustera.

2. Quelques algorithmes mathématiques en langage naturel

Le langage naturel, pour nous, est le français. Nous utiliserons seulement des mots simples ; le texte sera clair et bien structuré. Nous éviterons tout effet de style et toute fantaisie. La première qualité d'un texte mathématique est d'être clair et sans ambiguïté.

On va voir que l'algorithmique introduit des méthodes constructives dans la plupart des spécialités mathématiques et que tout ce qui relève d'un traitement automatisé relève de l'algorithmique.

Exemple 1 : Cercle circonscrit à un triangle ABC.

Les données sont trois points distincts non alignés A, B et C (formant un véritable triangle). Appliquons-leur l'algorithme suivant :

1. Tracer la médiatrice (D) de AB
2. Tracer la médiatrice (D') de AC
3. Appeler O le point d'intersection de (D) et (D')
4. Tracer le cercle (C) de centre O et de rayon OA

Le résultat de cet algorithme est (C) qui s'appelle cercle circonscrit au triangle ABC. En entrée, il y a trois points formant un triangle, en sortie un cercle.

Cet algorithme n'est pas une démonstration. On pourrait le transformer en démonstration (ou en définition) en ajoutant au bon endroit les commentaires suivants :

1. (D) et (D') se coupent parce que les points A, B et C ne sont pas alignés.

2. O se trouvant sur (D), $OA=OB$; de même, O se trouvant sur (D'), $OA=OC$, donc (C) passe par A, B et C.
3. Le centre de tout cercle passant par A et B se trouve nécessairement sur (D) ; de même, le centre de tout cercle passant par A et C se trouve nécessairement sur (D'). Un cercle passant par A, B et C est donc nécessairement centré en O. Par conséquent, (C) est le seul cercle à avoir cette propriété.

Il est donc facile d'insérer des algorithmes mathématiques dans un cours.

Exemple 2 : Médiane d'une série statistique

Définissons cette notion à l'aide d'un algorithme :

Les données sont une série statistique (une suite de nombres réels) notée x_1, \dots, x_n .

Voici l'algorithme en question :

1. Ranger les nombres x_1, \dots, x_n dans l'ordre croissant.
2. Appeler y_1, \dots, y_n la suite croissante obtenue.
3. Si n est pair, noté $n = 2p$, poser $m = \frac{y_p + y_{p+1}}{2}$
sinon, n étant noté $n = 2p+1$, poser $m = y_{p+1}$.
4. Afficher m.
5. // m s'appelle la médiane de la série statistique x_1, \dots, x_n .

La dernière ligne commençant par // n'est pas une instruction mais un commentaire. Les commentaires sont indispensables dans les algorithmes car ils permettent d'expliquer le rôle des instructions successives.

L'instruction 4 est une instruction de sortie. Il manque une instruction d'entrée qui aurait pu être :

0. Lire la suite x_1, \dots, x_n .

Cette définition est constructive mais manque de sens. C'est le cours lui donnera le sens manquant. On remarquera que :

- cet algorithme comporte une instruction conditionnelle : si n est pair, on fait ceci ; sinon, on fait cela,
- les entrées sont constituées par une série statistique,
- les sorties par un nombre.

Exemple 3 : *somme S des carrés des entiers de 1 à n*

On appliquera l'algorithme suivant :

1. Lire n .
2. Poser $S = 0$.
3. Pour i variant de 1 à n , faire $S = S + i^2$
4. Afficher S .

L'intérêt de cet algorithme muni de ses instructions d'entrée et de sortie est qu'il comporte un calcul itératif, de 1 en 1, dont le nombre de pas, à savoir n , est donné. En général, on appelle cela une boucle « pour ». Ce type d'itération est déjà un outil puissant de calcul.

Exemple 4 : *le crible d'Ératosthène*

Le crible d'Ératosthène fournit la liste des nombres premiers de 1 à n , pour tout entier $n \geq 3$, en exécutant le programme suivant :

1. Lire n .
2. Ecrire la liste des nombres entiers de 1 à n .

3. Encercler 1 et 2, supprimer les autres multiples de 2 figurant dans la liste.
4. Entourer d'un cercle le plus petit des nombres non encerclés restants, s'il y en a, et supprimer les autres multiples de ce nombre.
5. Répéter l'étape 4 jusqu'à ce qu'il n'y ait plus dans la liste restante de nombres non encerclés.
6. Afficher la liste restante.

Si l'on veut produire la liste des nombres premiers de 1 à 500 par exemple, il suffit d'affecter à n la valeur 500 et d'exécuter l'algorithme. Ce sera long.

On remarque que :

- cet algorithme muni de ses instructions d'entrée et de sortie ne précise pas combien de fois l'étape 4 sera itérée. C'est l'instruction 5 qui contrôlera les itérations et les arrêtera au bon moment. Autrement dit, cet algorithme s'auto-contrôle. Il s'agit d'un calcul itératif avec une fin de boucle conditionnelle, ce que l'on appellera une boucle « tant que » en général : tant qu'il reste dans la liste des nombres non encerclés, on continue les itérations.
- les entrées et sorties sont respectivement constituées par un nombre entier ≥ 3 et une liste de nombres entiers.

3. *L'algorithmique : vers une spécialité nouvelle*

Les exemples ci-dessus montrent que l'on a toujours fait de l'algorithmique en langage naturel, à l'aide d'une feuille de papier et d'un crayon. On peut dire que :

- un algorithme est une série d'instructions simples (des calculs, dans un sens étendu,

par exemple des calculs sur des chaînes de caractères, des constructions géométriques, des calculs numériques ou formels) ; ces instructions peuvent comprendre des instructions conditionnelles ou des boucles ; on les exécute pas à pas dans l'ordre de l'algorithme ;

- un algorithme possède une certaine autonomie ; il peut s'auto-contrôler grâce à des instructions de contrôle ;
- un algorithme n'est pas conçu pour un seul calcul ; il s'exécute après que l'on ait affecté une valeur à ses entrées ; l'ensemble des valeurs possibles des entrées est, lui, entièrement spécifié ; l'algorithme s'applique à chaque choix possible des entrées et produit en sortie un objet mathématique ;
- les sorties sont aussi d'un type entièrement spécifié ;
- en élevant le débat, on peut donc concevoir tout algorithme comme une fonction qui à un élément de l'ensemble des entrées possibles fait correspondre un élément de l'ensemble des sorties possibles.

On pourrait en rester là pour l'algorithmique en Seconde. On passerait malheureusement à côté de ce qui rend les mathématiques si importantes et même carrément indispensables, à notre époque, dans de nombreux domaines scientifiques et technologiques.

Il faut dire aux élèves que sans algorithmes :

- la CAO (conception assistée par ordinateur) disparaîtrait : on ne fabriquerait plus de nouveaux modèles d'avion, de voitures, de lunettes !
- on n'améliorerait plus les appareils électroniques, notamment les ordinateurs et les téléphones cellulaires !

- on n'aurait jamais envoyé une fusée sur la lune (pas de calcul de trajectoire) !
- on mourrait plus jeune (faute de traitement des images de scanner par exemple).

Ajoutons que l'on a de plus en plus besoin de nouveaux algorithmes, ce qui fait que cette spécialité (née des mathématiques mais que l'on peut aussi considérer maintenant comme une spécialité informatique) est en développement rapide (là-bas, en Inde).

Pour rendre possible cette formidable révolution, il aura fallu passer de l'algorithmique inconsciente d'autrefois à une algorithmique consciente, en dégageant les caractéristiques et les outils de cette spécialité naissante. Ce seul pas est en soi un progrès considérable. Le développement des calculateurs électroniques aura fait le reste.

Le projet de programme de mathématiques de Seconde indique quels sont ces caractéristiques et outils :

- instructions élémentaires (affectation, calcul, entrée, sortie)
- calcul itératif à nombre donné d'itérations
- calcul itératif à fin de boucle conditionnelle
- instruction conditionnelle.

Ils sont illustrés par les 4 exemples donnés précédemment.

II. — Intérêt pédagogique de l'algorithmique

Le but de l'enseignement d'algorithmique n'est pas d'appliquer des algorithmes, mais de résoudre des problèmes en créant des algorithmes. De ce point de vue, nos algorithmes n'ont rien à voir avec la cuisine ! Il est bien

sûr possible d'appliquer un algorithme de mathématiques sans le comprendre, mais ce cas de figure ne doit normalement pas se présenter et ne présente aucun intérêt.

- Un algorithme étudié comme solution d'un problème n'est pas une démonstration. C'est simplement une série d'instructions.
- Un algorithme ne doit jamais être réduit à un série d'instructions : il deviendrait très vite incompréhensible, le temps passant, pour son créateur (sans parler des utilisateurs) ; il doit contenir des commentaires explicatifs.
- Mettre au point un algorithme suppose qu'on sait le justifier. Une démonstration se cache toujours derrière un algorithme et cette démonstration sera, par la force des choses, bien structurée.
- Ce peut être une bonne pratique de donner à des élèves un algorithme et de leur demander de l'interpréter et de le justifier.

Nous allons bientôt passer de la feuille de papier et du crayon aux calculateurs électroniques.

Comme cette étape est attendue (évidemment) et parce que nous voulons continuer à parler de l'intérêt pédagogique de l'algorithmique, anticipons un peu.

- Un algorithme conçu en vue d'une application sur calculatrice ou ordinateur devrait toujours être rédigé d'abord en langage naturel parce que cela évacue les contraintes liées à ces matériels et logiciels et laisse donc l'esprit libre pour les mathématiques.
- Disons ensuite qu'ayant traduit un algorithme du langage naturel à un langage de programmation adapté au matériel et logiciel utilisés, il est très gratifiant de constater qu'il résout, quasi-instantanément, le

problème posé. Par exemple, personne n'aurait l'idée d'appliquer le crible d'Ératosthène à la main pour trouver la liste des nombres premiers inférieurs ou égaux à 1 000 000. Ce serait trop long. C'est possible grâce au calcul électronique.

- Malheureusement, il arrive qu'un algorithme mal conçu fonctionne et produise quelque chose qui n'est pas la solution du problème posé. Aussi le projet de programme de mathématiques de Seconde précise-t-il à juste titre qu'à l'occasion de l'écriture de programmes, il faudra donner aux élèves de bonnes habitudes de rigueur et un entraînement aux pratiques systématiques de vérification et de contrôle.
- On sait que la plupart des problèmes courants qui s'y prêtent ont déjà reçu leur solution algorithmique.
 - C'est l'occasion de répéter que l'apprentissage de l'algorithmique ne se réduit pas à l'utilisation de bibliothèques de programmes ou de fonctions pré-programmées. Cela devrait rester marginal en Seconde.
 - Par contre, apprendre comme autrefois à résoudre théoriquement un problème puis mettre en œuvre une solution algorithmique met incontestablement en valeur, et motive, l'étude théorique. Le programme met cela en exergue.
- Ajoutons que des tas de problèmes ne sont pas pourvus d'avance de solution toute prête : il faut programmer. Par exemple, on lance un dé. Le nombre i sort. On lance alors i dés et on appelle S la somme de ces i dés. Comment simuler S ?
- Un algorithme de calcul numérique n'est pas un algorithme de calcul formel. En calcul formel, on peut démontrer (vraie démonstration) que la somme des carrés des entiers

de 1 à n (voir l'exemple 3) est égale à $n(n + 1)/2$. À l'aide de l'algorithme de l'exemple 3, on pourrait constater que cette égalité est vérifiée pour quelques valeurs affectées à n . Mais, comme on ne peut pas donner à n toutes ses valeurs possibles, la formule est indémontrable de cette façon. C'est le genre de question que l'on peut poser aux élèves. Un logiciel de calcul formel traiterait n comme une variable et effectuerait les calculs algébriques adéquats constituant une démonstration.

- Finalement, l'algorithmique est passionnante, mais exigeante. Elle demande aux élèves des qualités nouvelles, ce qui laisse espérer qu'elle permettra de récupérer certains de ceux qui ne s'intéressent plus aux mathématiques, un peu comme le calcul des probabilités.
- Nous pensons qu'elle doit être introduite de manière extrêmement progressive en commençant par des problèmes très simples, voir [EO].

III. — Algorithmique, calculatrices et logiciels d'ordinateurs

1. Les divers types de langages

- Le problème est : comment rédiger un algorithme afin que la machine en comprenne les instructions. C'est une longue histoire. Elle a commencé à la naissance des ordinateurs et n'est pas finie.
- Distinguons seulement deux types de langages : les langages de programmation et les langages intermédiaires, qui se rapprochent du langage naturel. Les langages intermédiaires sont spécialisés, donc plus faciles à utiliser. Mais comme ils sont d'abord interprétés en langage de programmation, ils sont un peu plus lents, ce qui est relatif, vu la puissance des machines.

- La frontière entre langage de programmation et langage intermédiaire est par ailleurs floue car le vocabulaire et la syntaxe des seconds peuvent se rapprocher des premiers.

2. Importance du choix d'un langage

- En principe, ce choix n'est pas important. Mais on sait bien que pratiquement, il l'est énormément. Un des buts de la réunion était de se mettre d'accord sur un choix très limité de langages ou, mieux, de couples matériel-langage.
 - Il n'y a pas de réponse simple, d'autant plus que nous avons compliqué le problème en étendant le problème du choix d'un langage à l'enseignement de la géométrie et du calcul numérique et formel au lycée.
- Au contraire, en ce qui concerne les logiciels d'ordinateur (géométrie, géométrie dynamique, tableur, calcul numérique, calcul formel), nous nous sommes limités, aux « bons » logiciels gratuits et libres dont on peut penser qu'ils resteront gratuits et dont le développement n'a pas été abandonné. Nous omettons donc des logiciels commerciaux très connus.
 - Les langages de programmation utilisables librement avec tous les systèmes d'exploitation, donc dans tous les établissements et chez la plupart des élèves, sont nombreux.
- Tout environnement demande un apprentissage basique, que ce soit une calculatrice ou un logiciel. Cet apprentissage est rapide. On peut dire que les syntaxes des langages actuels s'acquièrent facilement.

3. Cas des calculatrices

Les calculatrices programmables sont pourvues d'un langage intermédiaire. Elles

suffisent pour l'algorithmique de Seconde ainsi que pour le calcul (numérique ou formel) du lycée.

- Elles permettent de se passer d'une salle informatique équipée, ce qui peut être très important.
- Les ordinateurs leur sont supérieurs seulement par leurs possibilités de connexion à des réseaux, leurs grands écrans et leur convivialité : utiliser des combinaisons de petites touches de calculatrice n'est pas agréable, pas plus que regarder un écran très petit, aux possibilités graphiques faibles.
- Par conséquent (cf. le premier item) et malgré ces réserves, on peut penser que les calculatrices resteront très utilisées. Certains d'entre nous pensent que c'est le meilleur choix ou parfois le seul choix possible, à court et moyen terme.
- Le professeur qui choisira d'utiliser exclusivement ou principalement une calculatrice pour le calcul (algorithmique en Seconde comprise) aura grandement déblayé son sujet. Il devra gérer l'hétérogénéité des machines, ce qui n'est pas simple.
- A ce sujet, il est bien sûr très souhaitable que le choix d'une calculatrice se fasse au niveau de l'établissement, en intégrant l'ensemble des cursus.
- On peut remarquer à ce sujet que le programme n'indique pas comment l'algorithmique sera prise en compte dans les classes ultérieures.

4. Cas des ordinateurs

- La dernière remarque ci-dessus invite à une certaine prudence quant au choix d'un environnement logiciel de programmation.
- Même si certains élèves n'en possèdent pas, l'ordinateur est quand même largement

répandu. De plus, des outils gratuits de programmation sont disponibles.

- XCas a des fonctionnalités nombreuses (calcul formel, tableur, géométrie dynamique, etc.) dans tous les environnements (en local Windows, Linux, MacOS, ... ou en ligne), que l'on n'est pas obligé d'employer toutes dans un premier temps. En limitant l'usage à son module de programmation, on écrira des algorithmes aussi simplement qu'avec d'autres outils, et l'on pourra profiter ultérieurement de ses possibilités plus avancées.
- Scilab est un logiciel de calcul très performant. Il est disponible sous Windows ou Linux sous une forme adaptée au lycée appelée scilab pour les lycées. Des commandes ont été francisées et surtout un document de présentation de 29 pages très aérées devrait suffire à l'utilisateur à ce niveau. C'est peut-être le logiciel de programmation qui demande l'investissement minimum sans hypothéquer des études scientifiques futures.
- On peut aussi se tourner vers des logiciels spécialisés (géométrie dynamique, tableur, calcul numérique, calcul formel).
- Le tableur (voir [EO]) permet de traiter certains algorithmes, mais il n'est pas fait pour les boucles/boucles conditionnelles. Pour cela, il faut le programmer en faisant appel à un langage de programmation supplémentaire, le Basic d'OOo, qui a l'avantage d'être embarqué dans la suite logicielle bureautique du même nom dont sont déjà familiers/pourvus les élèves/établissements.
- Les langages de programmation (Pascal, C, ...) sont des langages structurés fondamentaux. Ils s'ouvrent sur le développement logiciel autonome. Ils demandent une installation (simple) d'un environnement spécifique non mathématique. Parmi ces lan-

gages, citons le JavaScript, solution simple et passe-partout grâce à un éditeur léger utilisable en ligne ou hors-ligne. Il débouche sur la programmation de pages web.

- Pour plus de détails, nous renvoyons aux documents annexes [EO] et [AJ], dont nous recommandons la lecture.

5. En guise de conclusion

Risquons-nous à proposer, en dépit de toutes les réserves exprimées tout au long de ce « Point de vue » :

- *Les calculatrices programmables*
- *XCas et son module de programmation (en ligne et en local)*
- *Scilab pour les lycées*
- *JavaScript et son éditeur html léger (en ligne et en local)*
- *Basic intégré à la suite bureautique OOo (via le tableur par exemple)*

- *puis toute autre solution à la convenance du professeur.*

IV. — Publications de l'Irem de Lille en algorithmique pour la Seconde

- *Notre idée est de commencer par la mise en ligne de 4 ou 5 activités réellement basiques.*
- *Ensuite, nous pourrions mettre en ligne des activités donnant des solutions algorithmiques à des problèmes usuels.*
- *Pour remédier à la difficulté du choix des matériels et langages, nous essaierons toujours de donner une solution sur calculatrice et une au-moins une solution sur ordinateur.*
- *Rappelons que toute contribution est la bienvenue et que le site indique la marche à suivre pour proposer une activité.*