Chloé Ubéra
IREM d'Aquitaine

Philippe Lac
Alex Esbelin
IREM de Clermont-Ferrand

# Algorithmique entre programmation et démarche mathématique

## 1 Introduction

Une conception quasi unanimement acceptée attribue à l'algorithmique la seule fonction d'outil pour la programmation ; si c'est la seule, son introduction ne peut être naturelle que dans un projet de programmation. Or l'informatique est en dehors (à côté, mais en dehors) des mathématiques. L'introduction de l'algorithmique dans les programmes du secondaire relève donc de l'injonction institutionnelle et ne peut apparaître que plaquée sur l'enseignement des mathématiques, soit comme un antalgique pour calmer le manque de sciences du numérique, soit comme un accessoire "à la mode" destiné à rendre présentable un vieux vêtement. En l'absence d'une culture algorithmique commune aux enseignants de mathématiques, les modalités de son enseignement dépendent essentiellement de la formulation de cette injonction. La situation est bien différente par exemple pour la logique ou les nombres complexes.
Le propos de ce texte est de chercher à préciser les rapports entretenus par les mathématiques et l'informatique pour trouver à l'algorithmique une place qui soit pédagogiquement naturelle. Un tel travail n'aboutira que dans des décennies, quand les enseignants du secondaire auront collectivement construit une culture commune dans ce domaine.

## 2 Points de départ

### 2.1 Comment évaluer la production d'un algorithme par un élève ? Que veut-on que les élèves sachent ?

Le problème suivant est supposé être proposé à des élèves de lycée. Il suit une organisation pourrait être celle d'un problème de bac [1]. Contrairement à la partie "mathématique", seulement ébauchée, la partie "algorithmique" est donnée en détail. La question que nous envisagerons ne concerne pas le problème lui-même, mais des réponses (imaginaires) d'élèves qui lui font suite.

**Résumé de la première partie mathématique**
On considère la fonction $f$ définie sur $[-2;1]$ par $f(x) = 2x^3 - 3x^2 - 6x$. Le tableau de variations de $f$ sur $[-2;1]$ ci-contre est donné aux élèves ou déterminé par eux.

| $x$ | $-2$ | | $x_0$ | | $1$ |
|---|---|---|---|---|---|
| f | | | $f(x_0)$ | | |
| | | ↗ | | ↘ | |
| | $-16$ | | | | $-7$ |

On veut donner une valeur approchée de $x_0$.
[2]

1. La situation utilisée a été imaginée par Henri Rolland, de l'IREM d'Aix-Marsille dans un tout autre but.
2. La remarque faite par un participant sur la disponibilité d'une méthode alternative simple (détermination de la valeur qui annule la fonction dérivée polynomiale de degré 2) n'invalide pas ce problème : il n'est pas proposé comme situation d'apprentissage, et particulièrement comme situation permettant de montrer l'intérêt d'une démarche algorithmique. L'exploitation pédagogique de cette remarque n'entre pas dans le champ de ce travail.

**Partie algorithmique**
1) Prenons $a = -2$ et $b = 1$. On divise l'intervalle $[a; b]$ en trois parties de même longueur avec $u = -1$ et $v = 0$. Calculer $f(u)$ et $f(v)$. Ces résultats permettent-ils de donner un intervalle moins large que $[a; b]$ contenant $x_0$ ?
2) Reprendre la méthode initiée au 1) en remplaçant $[a; b]$ par l'intervalle trouvé à la question précédante et obtenir un nouvel encadrement de $x_0$.
3) Si on répète encore deux fois la méthode, quelle sera alors la longueur de l'encadrement de $x_0$ obtenu ?
4) En s'appuyant sur les résultats précédents, écrire un algorithme permettant de calculer un encadrement de $x_0$ à une précision donnée.

**Question 1 : évaluer les quatre exemples de réponses suivantes**

**Réponse 1**
Je pars de l'intervalle $[a; b]$.
Je le découpe en trois parties égales avec $u$ et $v$.
Je calcule $f(u)$ et $f(v)$.
si $f(u) < f(v)$, je remplace $a$ par $u$ ;
si $f(u) > f(v)$, je remplace $b$ par $v$ ;
si $f(u) = f(v)$, je remplace $a$ par $u$ et $b$ par $v$ ;
Je recommence jusqu'à ce que $a - b$ soit inférieur à la précision voulue.

**Réponse 2**
A, B, E et f sont donnés.
Tant que B-A $\geq$ E
    H $\leftarrow$ B-A/3
    U$\leftarrow$A+H
    V$\leftarrow$B-H
    Si f(U)$\leq$ f(V) alors A$\leftarrow$U
    Si f(U)$\geq$ f(V) alors B$\leftarrow$V
Afficher(A,B)

*****************************************************

**Réponse 3**
Je pars de l'intervalle $[a; b]$.
Je calcule $u = \frac{2a+b}{3}$ et $v = \frac{a+2b}{3}$.
tant que $u - v \geq e$ :
je calcule $f(u)$ et $f(v)$ ;
si $f(u) < f(v)$, je remplace $a$ par $u$ ;
si $f(u) > f(v)$, je remplace $b$ par $v$ ;
si $f(u) = f(v)$, je remplace $a$ par $u$ et $b$ par $v$.
Dès que $u - v < e$, j'écris $[u; v]$

**Réponse 4**
On découpe en trois parties égales l'intervalle $[a; b]$ qui sont $[a; u]$, $[u; v]$ et $[v; b]$. Si $f(u) < f(v)$, on remplace $a$ par $u$ ; si $f(u) > f(v)$, on remplace $b$ par $v$ ; sinon, on remplace $a$ par $u$ et $b$ par $v$. On recommence autant de fois que nécessaire.

Ces réponses peuvent paraître très différentes. Le plus souvent, la deuxième est valorisée. Toutes les quatre dénotent cependant une bonne compréhension du principe de l'algorithme. Dans la première, les instructions *Je ... découpe [l'intervalle] en trois parties égales avec u et v* et *Je recommence jusqu'à ce que a−b soit inférieur à la précision voulue* peuvent paraître imprécises. Mais la première de ces deux instructions définit précisément $u$ et $v$ et la seconde fournit le principe d'un test d'arrêt. Une discussion pourrait s'engager sur la pertinence de "inférieur" (strict ou large ?). Mais le cahier des charges sous-jacent à la question 4 n'est pas plus précis. La seule critique occasionnellement faite à la réponse 2 est que les variables ne sont pas typées, alors que l'opération "/3" concerne des nombres flottant et qu'un typage dynamique ou naturel attribuerait à A et B le type entier. Une telle critique met en évidence le fait que nous voulons mettre en avant dans notre conclusion provisoire :

**Conclusion provisoire :** *notre hypothèse (qui mériterait d'être approfondie) est que la majorité des enseignants évalueront la réponse 2 comme la meilleure, et que cette évaluation met en évidence la conception suivante :*

*La fonction des algorithmes est d'être des outils utiles pour la programmation*

## 2.2 Partons maintenant de la certitude suivante :

*Une fonction des algorithmes est d'être des outils utiles pour la programmation*

Cette affirmation relève de l'évidence si on accepte les deux points suivants :
⋆ ce n'est ni une définition, ni une caractérisation ;
⋆ ce n'est pas nécessairement la seule fonction.

*Supposons (provisoirement) l'hypothèse suivante : c'est la seule*

Sous cette hypothèse, on peut suspecter que l'introduction des algorithmes dans les programmes de mathématiques soit le cheval de Troie de l'informatique (au sens de sciences du numérique) qu'*on* chercherait à faire entrer dans la place de l'enseignement secondaire. Evisageons maintenant deux sous-hypothèses suivant que cette place lui serait (enfin) ouverte ou non :

• Cas 1 : *que pourrait-il se passer dans le cas où les programmes des lycées prévoiraient un enseignement informatique ?* Dans les départements d'informatique des IUT, la fonction envisagée ci-dessus pour l'algorithmique est naturellement le fil directeur de l'organisation de son enseignement. Voici un extrait des programmes :

Programme Pédagogique National
DUT Informatique 2013

⋆ Connaissances et Compétences « Informatique »
Algorithmique et Programmation :
Ces enseignements de tronc commun ... doivent permettre :
• d'acquérir les connaissances nécessaires à la réalisation de logiciels,
• d'acquérir les concepts transversaux aux différents champs de l'informatique en terme de raisonnement, d'abstraction et de mise en oeuvre de solutions,
• de développer des compétences permettant de comprendre, faire évoluer, d'assurer la maintenance et de déployer une application logicielle,
• d'apprendre à participer à un travail d'équipe en charge d'un projet et à être autonome dans la réalisation d'une mission.
⋆ Connaissances et Compétences « Générales »
Les mathématiques sont aujourd'hui un outil nécessaire à la compréhension des sciences et en particulier à l'informatique. On peut citer comme exemples, ceux reliés au domaine informatique, l'arithmétique pour la théorie de la cryptographie, l'algèbre linéaire pour la théorie du codage ; l'analyse et la géométrie pour le traitement des signaux et des images, les probabilités et les statistiques pour l'informatique de gestion et le traitement des données sans oublier les graphes, langages et les grammaires pour la théorie des langages. [a]

*a.* Dans la version intermédiaire 6.4, on trouvait :
**Champs disciplinaires, unités d'enseignements**
*Les champs disciplinaires de culture scientifique, sociale et humaine (environ50% des enseignements)*
Le champ disciplinaire « Mathématiques », support théorique des technologies de l'information et de la communication, apporte les connaissances reliées au domaine informatique : l'arithmétique pour la théorie de la cryptographie, ... *etc. à l'identique* ... sans oublier les graphes, langages et les grammaires pour la théorie des langages et l'étude des réseaux. Globalement, ces enseignements participent aussi au développement de l'aptitude à l'expression et à la communication scientifique ainsi que de l'aptitude à la formalisation et à la modélisation.
*Les champs disciplinaires "informatique" (environ 50% des enseignements)*
Le champ disciplinaire **"Algorithmique - Programmation - Langages"** couvre l'ensemble du spectre de l'activité de développement de logiciel. Outre la présentation des bases théoriques de la construction d'un programme (algorithmique, décomposition de problèmes en sous-problèmes, mécanismes de validation),...

On ne saurait être plus clair.

**Conclusion provisoire :** *notre hypothèse est que l'apparition de l'informatique chassera l'algorithmique du domaine officiel des mathématiques*

• Cas 2 : *que faire dans le cas où le cursus ne prévoit pas d'enseignement informatique* comme c'est le cas au lycée actuellement ? D 'abord, envisager une autre fonction de l'algorithmique, si il en est une.

# 3 Concepts de l'algorithmique et de sa didactique

## 3.1 Qu'est-ce qu'un algorithme ?

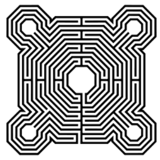
### 3.1.1 Un peu d'histoire

Entre les XIIème et XVème siècle, le mot désigne les techniques opératoires sur les entiers et les décimaux héritées des arabes (d'où justement le mot, issu du nom du mathématicien Al Khwarizmi, comme chacun le sait) par opposition à celles héritées des romains, pronées par les abacistes.
Bien que cette opposition soit encore aujourd'hui en débat à l'école primaire (à propos de la question suivante : quels sont les avantages respectifs de l'usage des calculatrices et de l'usage des techniques classiques dans les apprentissages des opérations), nous ne l'abordons pas (remarquer que la question ne se pose plus depuis des décennies en ce qui concerne l'opération d'extraction de racine carrée !).

Au XIXème et au début du XXème, le terme est utilisé dans le champ philosophique dans le cadre d'études sur les différentes formes de pensée [3].

Bien sûr, des algorithmes existent depuis bien avant ces deux périodes, mais le terme lui même ne semble pas avoir été utilisé avant le XX ème siècle dans d'autres acceptions que les deux citées ci-dessus. Il n'apparaît pas par exemple dans l'article de Tarry concernant le problème ci-dessous [4].

### 3.1.2 Le problème du labyrinthe :



Les gardes de Minos vous ont conduit à l'intérieur du labyrinthe et vous y abandonnent !
**Question 2 : Montrer qu'il est possible de sortir !**
**Démontrer le Théorème : il existe un chemin qui conduit de tout point d'un labyrinthe vers la sortie.**

Quelques réponses possibles :

**Réponse 1 :** suivre au retour le chemin de l'aller. [5]

**Réponse 2 :** suivre le fil d'Ariane.

**Réponse 3 :** suivre les déplacements suivi pour l'aller en inversant les directions et l'ordre.

**Réponse 4 :** la méthode de Tarry [6] :

> Tout labyrinthe peut être parcouru en une seule course, en passant deux fois en sens contraire par chacune des allées, sans qu'il soit nécessaire d'en connaître le plan.
> Pour résoudre ce problème, il suffit d'observer cette règle unique :
> Ne reprendre l'allée initiale qui a conduit à un carrefour pour la première fois que lorsqu'on ne peut pas faire autrement.
> Nous ferons d'abord quelques remarques.........
> En suivant cette marche pratique, un voyageur perdu dans un labyrinthe ou dans des catacombes, retrouvera forcément l'entrée avant d'avoir parcouru toutes les allées et sans passer plus de deux fois par la même allée.
> Ce qui démontre qu'un labyrinthe n'est jamais inextricable, et que le meilleur fil d'Ariane est le fil de raisonnement.

3. *Un algorithme conventionnel - qui d'ailleurs n'a de sens que rapporté au langage - n'exprimera jamais que la nature sans l'homme. Il n'y a donc pas à la rigueur de signes conventionnels, simple notation d'une pensée pure et claire pour elle-même, il n'y a que des paroles dans lesquelles se contracte l'histoire de toute une langue, et qui accomplissent la communication sans aucune garantie, au milieu d'incroyables hasards linguistiques.* M. Merleau-Ponty, Phénoménologie de la perception,1945, p. 219.
*Mais le cerveau, nous l'avons assez vu, possède une activité propre ; il peut opérer des constructions, à base d'images ou d'algorithmes, à l'aide desquelles il relie entre elles les sensations et les transforme en « objets ». Il est parfaitement vrai que la perception, conformément à la thèse de Lachelier et de M. Brunschvicg, suppose déjà une « réflexion », du même ordre que celle qui fait la science.* R. Ruyer, Esquisse d'une philosophie de la structure,1930, p. 218.

4. On peut trouver dans la littérature mathématque des XVIIIème et XIXème de nombreux autres exemples de ce type, chez Euler, Hamilton, ....

5. Comparer avec l'argument "le labyrinthe est modélisé par un graphe connexe et la définition de la connexité entraine la conclusion".

6. G. Tarry, "Le problème des labyrinthes", Nouvelles Annales de Mathématiques, ser. 3, 14 (1895), 187-190.

**Réponse 5 :** faire un intinéraire sur le plan.

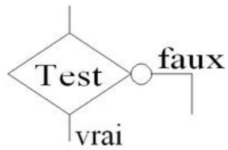
Nous pensons pouvoir mettre en évidence que si la description de ce que doit être un algorithme n'est pas facile, le consensus pour savoir parmi les méthodes précédentes lesquelles sont algorithmiques est facile à atteindre :
⋆ une méthode ne peut être retenue comme algorithmique si elle utilise une connaissance globale de la structure de donnée ou par la connaissance de la réponse qu'on cherche à fournir (réponses 1, 2 et 5) ;
⋆ la question des moyens disponibles nécessite un accord : si je peux mémoriser (d'une quelcoque manière) le chemin de l'aller, alors la réponse 3 fournit un algorithme ; quant à la méthode 4, elle fournit un algorithme sous réserve de pouvoir mémoriser les passages aux intersections.
Il apparaît ainsi une période de l'histoire des mathématiques pendant laquelle l'algorithme n'est pas un outil, encore moins un objet, mais relève d'une démarche de résolution de problème dont les contraintes font l'objet d'un consensus.

## 3.2 de la *notion* au *concept*[7]

Pour les informaticiens, la fonction de l'algorithmique est naturellement d'être un outil pour la programmation, l'objectif pédagogique étant d'encadrer la pratique de la rédaction des algorithmes pour la rendre plus efficace. On trouvera facilement de nombreuses variantes des *commandements* de Mac Naughton suivants,

- ⋆ Un algorithme doit pouvoir être écrit dans un langage de programmation ;
- ⋆ Son exécution dépend d'une donnée d'entrée et produit une donnée de sortie
- ⋆ Il exécute les unes après les autres un nombre fini étapes ;
- ⋆ Son déroulement à chaque état dépend exclusivement de son état.

On sait aussi que des "langages algorithmiques" ont été conçus pour encadrer très étroitement cette pratique. Par exemple par association de symboles comme celui ci-contre, qui représente une instruction conditionnelle.




Un enseignant de mathématiques ne sera pas étonné de l'existence de ces extrêmes, lui qui navigue entre deux extrêmes pour l'apprentissage de la résolution des équations par exemple.

Mais qu'en est-il des mathématiciens ? Lors de leur deuxième congrès international, à Paris en 1900, David Hilbert présente une liste de problèmes parmi lesquels *De la possibilité de résoudre une équation diophantienne. On donne une équation diophantienne à un nombre quelconque d'inconnues et à coefficients entiers rationnels : on doit trouver une méthode par laquelle, au moyen d'un nombre fini d'opérations, on pourra décider si l'équation est résoluble en nombres entiers rationnels*[8]

Pour en arriver à un théorème énonçant l'inexistence d'une telle méthode, il aura fallu donner un statut mathématique à la notion de méthode. C'est ce qu'a fait par exemple Turing en définissant ses fameuses machines, sous une forme comme suit :

> Une machine de Turing est un septuplet $(Q, \Gamma, B, \Sigma, q_0, \delta, F)$ où
> $Q$ est un ensemble fini d'états ;
> $\Gamma$ est l'alphabet de travail des symboles de la bande ;
> $B \in \Gamma$ est un symbole particulier (dit blanc) ;
> $\Sigma$ est l'alphabet des symboles en entrée ($\Sigma \subseteq \Gamma - \{B\}$) ;
> $q_0 \in Q$ est l'état initial ;
> $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ est la fonction de transition ;
> $F \subseteq Q$ est l'ensemble des états acceptants (ou finaux, terminaux)....

7. Cette distinction est utilisée par exemple dans Chabert et al. "histoire d'algorithmes"
8. *Entscheidung der Lösbarkeit einer diophantischen Gleichung. Eine diophantische Gleichung mit irgendwelchen Unbekannten und mit ganzen rationalen Zahlkoefficienten sei vorgelegt : man soll ein Verfahren angeben, nach welchen sich mittels einer endlichen Anzahl von Operationen entscheiden lässt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.*

Nous n'écrivons pas tout : il s'agit ici seulement de montrer qu'une telle définition appartient bien au champ des mathématiques.

**Conclusion provisoire :** *tant qu'il s'agit de produire des modes de résolution d'un problème (par exemple de résolution des équation polynomiales), il suffit d'un consensus pour décider de la correction de la* **méthode**. *La nécessité d'une* **définition** *n'apparaît que lorsqu'il s'agit de démontrer l'inexistence d'une méthode ou d'un procédé. Le concept d'algorithme en mathématiques à été élaboré dans un tel but.*

## 3.3 Y-a-t'il des concepts spécifiques à l'informatique ?

Alice veut s'identifier chaque fois qu'elle allume Bob son ordinateur.

Elle a donc sauvegardé un mot de passe, et chaque fois qu'elle allume son ordinateur, elle écrit son mot de passe et Bob vérifie en comparant avec celui qu'elle a sauvegardé.

Mais Charles le pirate peut lire le contenu de la mémoire. **Question : comment empêcher qu'il ne découvre puis utilise le mot de passe ?**

**Réponse :** on utilise une fonction $f$ difficilement inversible : la mémoire de l'ordinateur contient un algorithme de calcul de $f$ et l'image par $f$ du mot de passe $\mathbf{a}$ d'Alice. Quand quelqu'un veut s'identifier avec un mot de passe $\mathbf{b}$, Bob utilise l'agorithme pour calculer $f(\mathbf{b})$ et compare avec la liste des autorisations. Ce procédé est efficace si la connaissance de l'algorithme de calcul d'une image par $f$ et celle de $f(\mathbf{a})$ ne permet pas de déterminer $\mathbf{a}$ au moins dans des conditions satisfaisante (par exemple un temps de calcul suffisament court).

Alice veut pouvoir communiquer avec Bob à distance. Charles le pirate apprend comment intercepter les messages. **Question : Comment faire pour que les informations qu'il découvre ne lui permettent pas de se faire passer pour Alice ?**

**Question : comment deux interlocuteurs peuvent se mettre d'accord sur un code secret de telle manière qu'un piratage soit repéré ?**

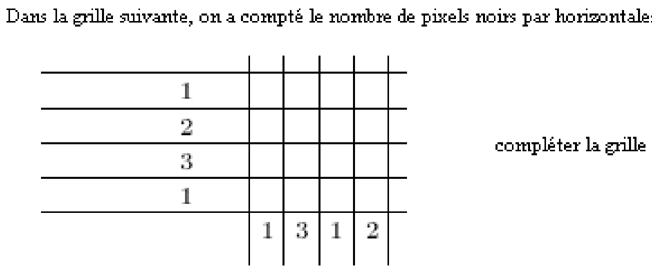
**Question : comment deux interlocuteurs peuvent se mettre d'accord sur un code secret par des échanges entièrement publics ?**

Pour la réponse à ces questions, il faudra se reporter à l'article qui développera ce travail !

**Conclusion provisoire :** *il existe des concepts spécifiques à l'informatique : parmi ceux-ci, celui de* **fonction difficilement inversible**, *qui utilise les concepts de* **rapidité d'exécution d'un programme** *et de* **complexité en temps d'un algorithme**.

### 3.4 Y-a-t'il des concepts partagés entre mathématique et informatique ?

On veut commencer à préciser ici les relations entre l'algorithmique et (de part et d'autre) les sciences du numérique et les mathématiques, (en particulier spécifier autant que possible les problèmes, concepts et méthodes relevant de chacune de ces disciplines). Euh... en fait surtout du côté des mathématiques. Hors de tout contexte d'enseignement, la tomographie médicale conduit à un problème de tomographie discrète comme le suivant




Une fois le problème résolu dans le cas de figure proposé, (et même avant), on se convainc aisément qu'un algorithme exhaustif, qui envisage toutes les grilles possibles et vérifie si leurs projections tomographiques conviennent, permet de résoudre ce problème.
**Question : comment représenter toutes les grilles ?**

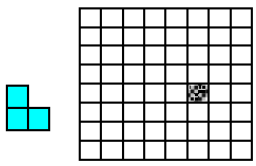
**Réponse :** une façon de représenter ces grilles consiste à représenter chacune par un entier écrit en binaire, de longueur 16, chaque chiffre représentant la coloration noir/blanc de chacune des cases : par exemple 1000000000001000 représente la grille ayant un pixel noir en haut à gauche et un pixel noir en bas à gauche...

**Conclusion :** *l'identification et le codage d'une structure de donnée est un exemple de problème partagé entre informatique et mathématique ; plus généralement aussi, la représentation des structures finies. La difficulté pour un mathématicien de répondre spontanément à cette question montre la spécificité de certains concepts des sciences du numérique.*

### 3.5 Démarche algorithmique

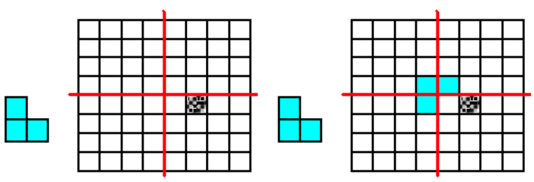
#### 3.5.1 Et tu recommences

**Question : résoudre le problème suivant :**
Etant donné un carré de taille $2^n$, $n$ quelconque avec un « trou » d'une case, pour quelles positions du trou est-il pavable par des triminos coudés ? Ci-contre, un cas particulier [a]



a. D'après D. Grenier et C. Payan ; voir par ex *Deux exemples de situations de recherche pour la classe*

**Réponse :**



On découpe le carré en quatre carrés de même taille. En posant un trimino sur les trois petits carrés qui ne comportent pas le trou, on obtient quatre petits carrés comportant un trou. Ansi, si il est possible de résoudre le problème pour des carrés de taille $2^n$, il est possible de le résoudre le problème pour des carrés de taille $2^{n+1}$.

L'algorithme évoqué ci-dessus réoud le problème (démontre le théorème) de manière complète et acceptable par un mathématicien. Pourtant, la réalisation d'un programme requiert une technicité assez importante.

**Conclusion :** *une démarche algorithmique peut se développer dans un cadre mathématique sans qu'il soit nécessaire de recourir à l'informatique pour la justifier, l'illustrer, l'institutionnaliser, ....*

C'est la méthode *diviser pour régner* : pour ranger un paquet de copies dans l'ordre alphabétique, j'en prend quatre que je range dans l'ordre alphabétique, puis quatre autres, puis je fusionne les deux tas. Et ainsi de suite...

Les deux autres éléments clés de la démarche algorithmique que nous avons identifiés à ce jour sont les algorithmes exhaustifs et les recherches par balayage.

### 3.5.2 Algorithmes exhaustifs

Les algorithmes cités au début de la partie précédente sont tous des algorithmes exhaustifs. Un algorithme exhaustif permet la résolution d'un problème de recherche d'informations sur la partie d'un ensemble donné définie par une contrainte sur les éléments de cet ensemble. Il est possible de le décliner sous des variantes adaptées à l'information cherchée :

⋆ existe-t'il des éléments $x$ de $X$ satisfaisant $P(x)$ ?

> Pour chaque élément $x$ de $X$, tester si $P(x)$ ;
> Dès qu'un test est positif, répondre "il existe des éléments $x$ de $X$ satisfaisant $P(x)$" ;
> Si aucun test n'est positif, répondre le contraire.

⋆ combien d'éléments $x$ de $X$ satisfont $P(x)$ ?

> Pour chaque élément $x$ de $X$, tester si $P(x)$ ;
> Quand un test est positif, incrémenter le compteur des réponses de 1 ;

La question à envisager ici est la suivante :

**Rédiger un algorithme exhaustif relève-t'il des mathématiques ?**

On pourrait d'abord considérer la question de savoir ce qu'est *rédiger un algorithme exhaustif* ? Si nous nous sommes bien compris nous même, il ne reste plus qu'à

⋆ écrire une fonction qui teste si $P(x)$ ;

⋆ énumérer tous les éléments de $X$.

D'un point de vue mathématique, ceci ne pose pas de problème. En revanche, l'énumération effective des objets considérés, dans un projet de programmation, peut être difficile : on a vu de telles structures de données plus haut. On peut aussi penser à *énumérer tous les graphes à 12 sommets*, *énumérer tous les arbres bianires de profondeur 6*, ...

### 3.5.3 Systématiser le tâtonnement

C'est la méthode du juste prix, du balayage, de la dichotomie. Le problème "trichotomie" présenté ci-dessus en relève.

# 4 Conclusion

Il est trop tôt pour être péremptoire. Essayons cependant un critère permettant de distinguer un algorithme et un programme :
⋆ un programme est un texte destiné à être exécuté par un ordinateur ;
⋆ un algorithme est un texte destiné à être compris par un être humain.
Il ne semble pas inenvisageable d'attribuer aux algorithmes non seulement la fonction d'outil pour la rédaction d'un programme, mais aussi celle d'outil pour comprendre et concevoir des méthodes de résolution de problème. Il apparaîtra ainsi qu'au jour tant attendu où les sciences du numérique auront enfin une place dans l'enseignement secondaire, la culture que les enseignants de mathématiques auront construite puisse être adaptée dans une synergie qui évite de renvoyer dos à dos les deux disciplines (et de renvoyer les mathématiques au statut de discipline pour la culture générale).