Quelques propositions d'algorithmes

Sont indiquées entre des crochets une ou plusieurs propositions de logiciels à utiliser pour mettre en œuvre l'algorithme proposé ainsi que des étoiles indiquant sa difficulté de réalisation. Cette difficulté est bien sûr une appréciation personnelle purement subjective. Les algorithmes réalisables avec AlgoBox le sont également avec Scratch, Python, Xcas, Scilab ou tout autre logiciel de programmation pour peu qu'on en maitrise le langage. La plupart sont aussi réalisables avec un tableur, Excel ou OpenOffice, mais ne présentent pas toujours le même intérêt (par exemple la présence de la fonction ALEA.ENTRE.BORNES rend inintéressant l'item 1). La "philosophie algorithmique" des tableurs est parfois assez différente de celle des logiciels de programmation. Mais ce sont des outils extrêmement puissants et souples qu'il serait, à mon avis, regrettable de négliger. Enfin les algorithmes les plus simples peuvent être réalisés sans difficulté sur une calculatrice programmable courante (TI 82 ou Graph 35).

1. **Tirage d'un nombre entier compris entre deux valeurs** [AlgoBox *]
   Créer un générateur de nombres entiers pseudo-aléatoires compris entre deux bornes à partir du générateur de nombres décimaux pseudo-aléatoires compris entre 0 et 1.

2. **Tirage sans remise de deux valeurs** [AlgoBox *]
   Désigner deux élèves au hasard dans une classe de 35 (tirer deux nombres distincts entre 1 et 35). On utilisera le générateur de nombres pseudo-aléatoires défini à l'item 1. (On peut élargir facilement à deux nombres compris entre 1 et *n*.)

3. **Tirage du Loto** [AlgoBox ***]
   Propose un tirage pseudo-aléatoire de six nombres, plus un, parmi 49 sans remise.

4. **Permutation de *n* éléments** [AlgoBox ***]
   C'est à la fois une généralisation (*n* éléments au lieu de 49) de l'algorithme précédent et un cas particulier (*n* éléments parmi *n*).

5. **Lancers de dés** [AlgoBox **]
   On utilise un dé à six faces (généralisation possible à *k* faces).
   Première version : On effectue *n* lancers et on affiche les fréquences obtenues.
   Deuxième version : On effectue *p* séries de *n* lancers et on affiche le tableau des fréquences obtenues.

6. **Ecriture décimale illimitée périodique d'un rationnel. (Division à virgule)** [AlgoBox **] [Excel *]
   Poursuivre une division aussi loin que nécessaire pout déterminer la période de l'écriture décimale illimitée d'un nombre rationnel.

7. **Détermination des racines d'une équation polynomiale par dichotomie** [AlgoBox ***] [Excel ***]
   On se propose de déterminer les zéros du polynôme définie sur $\mathbb{R}$ par :
   $$F_1(x) = x^6 + x^5 - 11x^4 - 10x^3 + 21x^2 + 9x - 3$$
   Ce polynôme s'annule pour six valeurs comprises entre -3 et 3. AlgoBox permet d'en déterminer des valeurs approchées avec une précision de 10^{-7}. ($2^{10} = 1024$ est voisin de 10^3, on gagne 3 décimales toutes les dix itérations.)

8. **Distance de deux points, milieu d'un segment, équation d'une droite passant par deux points, médiatrice d'un segment** [AlgoBox **]
   Connaissant les coordonnées de deux points du plan, on veut déterminer les coordonnées du milieu du segment ayant pour extrémités ces deux points, l'équation de la droite passant par ces deux points ainsi que l'équation de la médiatrice du segment. Cet algorithme peut être réalisé en plusieurs étapes. On peut également en faire une représentation graphique

9. **Avec trois points** [AlgoBox ***]
   Déterminer les longueurs des côtés, tester une condition d'alignement, déterminer les équations des droites, des médiatrices des côtés, déterminer les coordonnées du centre du cercle circonscrit ainsi que son rayon. Tracer le triangle, les 3 médiatrices et le cercle circonscrit. On peut imaginer encore bien d'autres choses (détermination de la nature du triangle, distance d'un point à la droite passant par les deux autres, détermination des médianes, des hauteurs, des bissectrices, enfin tout sur le triangle et plus encore).

10. Algorithme de Prabhakar [Excel **]

$u_0 \in \mathbb{N}$. $\left(u_k = \overline{a_n a_{n-1} ... a_2 a_1 a_0} = \sum_{i=0}^{n} \left(a_i \times 10^i \right) \right) \Rightarrow \left(u_{k+1} = \sum_{i=0}^{n} a_i^2 \right)$. (écriture pour profs de maths !)

Que constate-t-on ?

Quel que soit l'entier naturel duquel on part on aboutit soit au cycle

$4 \to 16 \to 37 \to 58 \to 89 \to 145 \to 42 \to 20 \to 4$ etc. soit à 1, soit à 0 (uniquement pour 0).

On peut étudier ce qu'il advient pour $p > 2$ en posant $u_{k+1} = \sum_{i=0}^{n} a_i^p$.

11. Recherche des extrema d'une fonction sur un intervalle [AlgoBox **]

Recherche du maximum et de minimum d'une fonction par balayage.

Par exemple, étude des extrema de la fonction *f* définie sur $\mathbb{R}$ par $f(x) = x^3 + x^2 - 3x - 1$.

12. Méthode de Monte-Carlo [AlgoBox *] [Excel *]

On tire de façon aléatoire, deux nombres *x* et *y*, compris entre 0 et 1 et on place dans le plan (rapporté à un repère orthonormal) le point *M* de coordonnées (*x* ; *y*). Il s'agit de faire apparaître la fréquence des points dont la distance à l'origine est strictement inférieure à 1. Cette fréquence tend vers $\pi/4$ mais il faut faire un grand nombre de tirages pour obtenir une approximation significative.

13. Les aiguilles de Buffon [Excel **]

On lance un grand nombre d'aiguilles sur un parquet formé de planches parallèles. On supposera que la largeur des planches est égale à la longueur des aiguilles. On compte le nombre de fois où les aiguilles tombent à cheval sur une rainure du parquet. Le rapport de ce nombre sur le nombre total de lancers tend vers $2/\pi$. Comme dans la méthode de Monte-Carlo la convergence est lente.

14. Tri par sélection [AlgoBox **]

On cherche dans la liste la plus petite valeur. On la place au début et on recommence.

Performance : 5000 valeurs triées en 1 min 55 s. [Pour un ordinateur de référence.]

15. Tri par insertion [AlgoBox **]

On prend une nouvelle valeur et on l'insère dans la liste déjà triée.

Performance : 5000 valeurs triées en 1 min 6 s. [Pour un ordinateur de référence.]

16. Tri par dénombrement [AlgoBox **]

On compte le nombre de tirages pour chacune des valeurs et on restitue ces valeurs dans l'ordre.

Performance : 5000 valeurs triées en 46 s. [Pour un ordinateur de référence.]

17. Algorithme de Babylone (ou algorithme de Héron) [AlgoBox **]

On cherche une valeur approchée de $\sqrt{a}$ pour $a \in \mathbb{R}_+^*$. On crée la suite suivante :

$u_0 = \frac{a}{2}$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = \frac{1}{2}\left(u_n + \frac{a}{u_n} \right)$. Cet algorithme converge très rapidement et il est facile à comprendre. On peut l'adapter sans difficulté pour rechercher la racine cubique ou la racine $n^{\text{ième}}$.

18. Algorithme de Kaprekar (mathématicien indien 1905-1988) [Excel ***]

On part d'un nombre u_0 écrit en base 10. v_1 est le nombre écrit à l'aide des mêmes chiffres pris dans l'ordre décroissant. w_1 est le nombre écrit avec les même chiffres pris dans l'ordre croissant. Enfin $u_1 = v_1 - w_1$, et on recommence.

<u>*Exemple*</u> : $u_0 = 351\,947$, $v_1 = 975\,431$, $w_1 = 134\,579$ et $u_1 = 975\,431 - 134\,579 = 840\,852$.

On observe que la suite aboutit rapidement à un point fixe (appelé "puits") :

$323\,980 \to 959\,931 \to 863\,832 \to 631\,764 \to 631\,764 \ldots$

ou la suite aboutit à un "cycle" :

$925\,168 \to 860\,832 \to 862\,632 \to 642\,654 \to 420\,876 \to 851\,742 \to 750\,843 \to 840\,852 \to 860\,832 \ldots$

19. **Fractions** [AlgoBox **]

On dispose des chiffres de 1 à 9. On forme un nombre *a* en choisissant quatre d'entre eux et un nombre *b* avec les 5 restants. Est-il possible que la fractions $\frac{a}{b}$ soit égale à $\frac{1}{2}$? $\frac{1}{3},\frac{1}{4},\frac{1}{5},\frac{1}{6},\frac{1}{7},\frac{1}{8},\frac{1}{9}$?

Il existe bien des manières de chercher les solutions à l'aide d'un algorithme mais une manière originale consiste à essayer des permutations (item 4) de façon aléatoires avec un test d'arrêt.

Ce qui revient à écrire n'importe quoi et compter sur la chance … et ça marche (le plus souvent) !

20. **Coder et décoder en "Jules César"** [AlgoBox **]

Il s'agit du codage le plus simple, celui par décalage des caractères.

Il parait naturel à un professeur de maths d'utiliser les congruences modulo 26 mais on peut aussi, moins savament, utiliser une structure alternative. Pour information les codes Ascii des minuscules vont de 97 à 122 ceux des majuscules de 65 à 90 et l'espacement a le code 32.

21. **La Bibliothèque de Babel** [AlgoBox **]

Dans la nouvelle *La Bibliothèque de Babel* (extraite de *Fictions*) l'écrivain argentin Jorge Luis Borges imagine une bibliothèque contenant tous les volumes possibles obtenus par combinaisons de vingt-cinq caractères.

« …chaque livre a quatre cent dix pages ; chaque page, quarante lignes, et chaque ligne, environ quatre-vingts caractères noirs. »

« Le manuscrit original du présent manuscrit ne contient ni chiffres ni majuscules. La ponctuation a été limitée à la virgule et au point. Ces deux signes, l'espace et les vingt-deux lettres de l'alphabet sont vingt-cinq symboles suffisants énumérés par l'inconnu. »

On veut écrire un algorithme qui génère une ou plusieurs pages d'un des volumes de cette bibliothèque.

Un calcul simple montre qu'il y a environ $25^{1\,312\,000} \approx 2\times10^{1\,834\,097}$ volumes possibles.

22. **Conjecture de Syracuse** [AlgoBox *] [Excel **]

$u_0 \in \mathbb{N}^*$. Pour tout entier naturel *k*, si u_k est impair alors $u_{k+1} = 3u_k + 1$, sinon $u_{k+1} = \frac{1}{2}u_k$. Il semblerait que quel que soit l'entier naturel u_0 cette suite aboutisse à 1. On obtient ensuite le cycle 4 – 2 – 1 indéfiniment. Cette conjecture est connu sous le nom de conjecture de Syracuse ou conjecture de Collatz ou conjecture d'Ulam ou conjecture tchèque. En dépit de la simplicité de son énoncé, elle n'a pas encore été démontrée. Le mathématicien hongrois Paul Erdös dit à son propos que « les mathématiques ne sont pas encore prêtes pour de tels problèmes » . Il ne s'agit bien sûr pas de la démontrer, mais on peut la découvrir, l'essayer, compter le nombre d'étapes, faire des statistiques, des tableaux.

23. **Intersection de deux droites** [AlgoBox *] [Excel *]

Les deux droites sont données par leurs équations cartésiennes réduites $(y = mx + p)$. Il s'agit en fonction des coefficients directeurs et des ordonnées à l'origine de déterminer si les droites sont confondues, disjointes ou sécantes. Dans ce dernier cas on pourra déterminer les coordonnées de leur point d'intersection. On peut ajouter éventuellement une représentation graphique.

24. **Image d'un réel par une fonction, tableau de valeurs, tracé** [AlgoBox *]

Il s'agit d'un algorithme d'initiation que l'on peut enrichir progressivement.

Déterminer l'image d'un nombre par une fonction numérique d'une variable réelle.

Faire un tableau de valeurs et une représentation graphique sur un intervalle.

		AlgoBox												Tableur	
				Types de			Structures			Opérations		Modules			
		Pertinence	Difficulté	NOMBRE	CHAINE	LISTE	SI … ALORS (… SINON)	POUR … DE … A	TANT QUE …	Nb. pseudo-aléatoires	Opérations sur les chaînes autres que la concaténation	Fonction	Repère	Pertinence	Difficulté
1	Tirage entre bornes	oui	*	oui			pos	pos	pos	oui					*
2	Tirage sans remise	oui	*	oui			oui			oui				oui	*
3	Loto	oui	***	oui	oui	oui		oui		oui				oui	***
4	Permutation	oui	***	oui		oui		oui		oui				oui	***
5	Dés	oui	**	oui	oui	oui		oui		oui				oui	**
6	Division à virgule	oui	**	oui	oui			oui	pos					oui	*
7	Dichotomie	oui	***	oui			oui		oui			oui	oui	oui	***
8	Deux points	oui	**	oui	oui		oui						oui	oui	**
9	Trois points	oui	***	oui	oui		oui	pos					oui	oui	***
10	Prabhakar													oui	**
11	Extrema	oui	**	oui	oui		oui	oui				oui	oui	oui	**
12	Monte-Carlo	oui	*	oui	oui		oui	oui						oui	*
13	Aiguilles de Buffon													oui	**
14	Tri par sélection	oui	**	oui		oui		oui		oui					
15	Tri par insertion	oui	**	oui		oui		oui	oui	oui					
16	Tri par dénombrement	oui	**	oui		oui	oui	oui		oui					
17	Babylone	oui	**	oui	oui				oui					oui	**
18	Kaprekar													oui	***
19	Fractions	oui	**	oui	oui	oui		oui	oui	oui				oui	**
20	Jules César	oui	**	oui	oui	oui	oui	oui	oui		oui			oui	**
21	Bibliothèque de Babel	oui	**	oui	oui		oui	oui			oui				
22	Syracuse	oui	*	oui					oui					oui	**
23	Intersections	oui	*	oui			oui						oui	oui	*
24	Fonctions	oui	*	oui	pos			pos				oui	oui	oui	*

La pertinence et la difficulté indiquées sur ce tableau ne sont qu'un avis.
De même dans l'utilisation des types de variables ou des structures, d'autres choix sont possibles.