

1. PGCD

Écrire une version itérative et une version récursive de l'algorithme permettant de calculer le *pgcd* de deux nombres entiers.

(a) Par l'algorithme d'Euclide

Principe :

Si b ne divise pas a , il faut remplacer b par le reste de la division de b par a et recommencer.
Sinon (b divise a), le *pgcd* est égal à b .

Exemple :

pgcd de 45 et 20 :

45 ne divise pas 20, $45 = 20 \times 2 + 5$

pgcd de 45 et 5

45 divise 5

pgcd : 5

(b) Par soustractions successives

Principe :

Si $a = b$, le *pgcd* est a . Sinon on calcule la différence entre le plus grand et le plus petit, et on calcule le *pgcd* de ce nombre avec le plus petit.

Exemple :

pgcd de 45 et 20 :

Différence : 25

pgcd de 25 et 20 :

Différence : 5

pgcd de 20 et 5

Différence : 15

pgcd de 15 et 5

Différence : 10

pgcd de 10 et 5

Différence : 5

pgcd : 5

2. Division par soustractions successives

Pour effectuer la division euclidienne de a par b , on construit une suite strictement décroissante $(a_i)_{i \in I}$ définie par une relation de récurrence : $a_0 = a$, $a_{i+1} = a_i - b$. Il existe un plus petit entier q tel que $a_q < b$. Le quotient de la division cherchée est q , et le reste $a - qb$.

Écrire l'algorithme de la division par soustractions successives (encore appelée algorithme d'Euclide).

- (a) En utilisant une boucle *tant que*
- (b) En utilisant une boucle *répéter jusqu'à*
- (c) En utilisant une fonction récursive

Dire quels sont les avantages et les inconvénients de chacune des méthodes.

3. Multiplication

Écrire l'algorithme de la multiplication alexandrine (d'Hypatique) dont le principe est le suivant :

Pour multiplier deux entiers X et Y, on détermine la suite :

$$\begin{aligned}(x_0, y_0) &= (x, y) \\ (x_{n+1}, y_{n+1}) &= (2x_n, y_n/2) \text{ (} / \text{ est une division entière)}\end{aligned}$$

Le produit XY est égal à la somme des x_i pour lesquels y_i est impair.

- (a) Justifier l'intérêt d'un tel algorithme pour un usage informatique
- (b) Écrire l'algorithme en utilisant une boucle *tant que*
- (c) Écrire l'algorithme en utilisant une fonction récursive

4. Comptage Shadok¹

Les shadoks² n'ont que quatre cases dans le cerveau, donc ne connaissent que quatre mots : GA BU ZO et MEU.

Quand il n'y a pas de shadok, on dit GA.

Quand il y en a un, on dit BU

Quand on en ajoute un, on dit ZO

Quand on en ajoute un, il y en a MEU

Quand on en ajoute un, ça dépasse la capacité du cerveau shadok, alors on les met dans une poubelle et on dit qu'il y a BU poubelle et GA shadok, puis BU poubelle et BU shadok, puis BU poubelle et ZO shadoks, puis BU poubelle et MEU shadoks. En ajoutant un shadok de plus, on remplit une poubelle de plus, ce qui fait ZO poubelles et GA shadok et ainsi de suite.

Arrivé à MEU poubelles et MEU shadoks, en ajoutant un shadok de plus, on remplit une nouvelle poubelle, mais on arrive à une poubelle de trop pour le cerveau shadok. On met donc les poubelles dans une grande poubelle et on a donc BU poubelle de poubelles, GA poubelle et GA shadok...

etc. etc.

- (a) Dire quel système de numération est utilisé par les Shadoks.
- (b) Si l'on dit qu'une poubelle est d'ordre 1, qu'une poubelle de poubelles est d'ordre 2, qu'une poubelle de poubelles de poubelles est d'ordre 3, etc. Combien de shadoks contient une poubelle d'ordre k ?

1. Voir la série télé animée des années 68 de Jacques Rouxel

2. Ce texte est extrait de la même série, je n'y suis donc pour rien.

- (c) Quel nombre est désigné par ZO-MEU-ZO-BU-ZO-GA ?
- (d) Ecrire le nombre 258 en numération Shadok.
- (e) Ecrire un algorithme permettant d'écrire un nombre quelconque en numération Shadok.

— Méthode : On cherchera à effectuer une série de divisions euclidiennes.

Rappel : Théorème

Soit a et b deux entiers naturels, avec b non nul, il existe un couple unique d'entiers naturels (q, r) tel que

$$a = bq + r \text{ et } 0 \leq r < b$$

L'entier a est appelé le dividende de cette division, b le diviseur, q le quotient, et r le reste.

Pour écrire un nombre en base b

On a :

$$N = bQ_0 + r_0, Q_0 = bQ_1 + r_1, Q_1 = bQ_2 + r_2, \dots, Q_{n-2} = bQ_{n-1} + r_{n-1}, Q_{n-1} = b \times 0 + r_n$$

$$N = bQ_0 + r_0 = b(bQ_1 + r_1) + r_0 = b^2Q_1 + br_1 + r_0 = b^2(bQ_2 + r_2) + br_1 + r_0 = b^3Q_2 + b^2r_2 + br_1 + r_0$$

5. Multiplication Shadok

Écrire l'algorithme de la multiplication sur une base 4, dont le principe est le suivant :

Pour multiplier deux entiers X et Y , on détermine la suite :

$$(x_0, y_0) = (x, y)$$

$$(x_{n+1}, y_{n+1}) = (4x_n, y_n/4) \text{ (} / \text{ est une division entière)}$$

Le produit XY est égal à la somme des x_i multiplié par y_i modulo 4

Est-ce que cela n'évoque pas quelque chose que vous avez déjà appris ?

6. Jouet

Un fabricant de jouets électroniques demande une étude informatique pour un rébus. Sur ce jouet, l'enfant dispose d'un ensemble de cartes en plastique sur lesquelles est imprimé un objet correspondant à un son unique : "pie", "haie",

"oeufs", etc. Pour constituer un mot, l'enfant devra disposer une suite de cartes dans des trous dans le bon ordre. La succession des sons correspondant à la prononciation du mot proposé. Bref un rébus.

Chaque carte en plastique est identifiée et reconnue par le jouet grâce à des picots qui appuient sur des interrupteurs.

— Combien de picots faut-il pour reconnaître 87 cartes différentes ?

— Est-il possible d'augmenter le nombre de cartes du jeu sans augmenter le nombre de picots ? De combien ?

— L'usine fait une grosse économie en enlevant un picot par carte. Combien de cartes différentes peut-on distinguer en faisant cette économie ?

— Le mot le plus long est fixé à 7 cartes. Combien de bits faut-il pour encoder un mot avec la réponse précédente ?

- Pour des raisons de fabrication, le client demande d'avoir un nombre de bits qui correspond à un multiple d'octets. Combien de bits doit-on retenir pour des mots de 7 lettres avec approximativement 87 cartes ?
- Combien de mots différents pourront être enregistrés en tout par l'appareil ?

7. Est-ce raisonnable ?

Une équipe marketing d'une grande enseigne de distribution cherche à construire des profils client. Pour cela, elle cherche à associer les 5 propriétés les plus courantes qui sont communes aux différentes marchandises récemment achetées par le même client. L'ensemble de ces propriétés constitue un *profil* abstrait qui servira à proposer des promotions sur des marchandises. L'algorithme qu'ils ont écrit est basé sur l'idée suivante :

Chaque client est listé et pour chaque marchandise de l'enseigne on regarde si elle est dans les achats récents du client. Puis à chacune de ces marchandises, on regarde l'ensemble des autres marchandises et on cherche d'une part un lien de correspondance entre les deux, et d'autre part si elle fait aussi partie des achats récents du client.

À chaque fois qu'une propriété est ainsi identifiée, on incrémente un compteur associé.

On prend enfin les 5 propriétés les mieux notées pour le client.

- Est-ce que cet algorithme est faux ?
- Écrire de façon plus économique cet algorithme

8. Comptage

Soit le tableau suivant :

Exemple de matériel de stockage	n	10^n	préfixe
	1000	10^3	kilo
	1000 000	10^6	méga
	1000 000 000	10^9	giga
	1000 000 000 000	10^{12}	téra
	1000 000 000 000 000	10^{15}	péta
	1000 000 000 000 000 000	10^{18}	exa
	1000 000 000 000 000 000 000	10^{21}	zetta
	1000 000 000 000 000 000 000 000	10^{24}	yotta

- (a) Les quantités désignent un nombre d'octets. Remplir la première colonne en trouvant des exemples historiques ou contemporains de stockage.
- (b) Donner des exemples d'application réclamant ces quantités de stockage (audio, vidéo, textes, etc.).

Soit un ensemble d'algorithmes dont la complexité en temps est exprimée par la première colonne du tableau suivant :

n	1	10	100	1000
C	C	C	C	C
$\log_2 n$	0	3,32	6,64	9,96
$n \log_2 n$	0	33,21	664,38	9965,78
n^2	1	100	10^3	10^6
n^3	1	1000	10^6	10^9
2^n	2	1024	10^{30}	10^{301}

- (a) Dessiner approximativement les graphes de ces fonctions
- (b) Rayer les cases où la complexité est rédhibitoire pour un usage informatique

9. Achetez des ordinateurs plus rapides !

Voici une fiction dans le domaine du génie logiciel.

Un logiciel est développé par une société de services pour une banque. Il est destiné à faire des recherches d'anomalies comptables sur plus de 500 000 comptes.

Le logiciel a été testé sur 100 comptes avec des test unitaires. Les expériences montrent qu'il est correct mais devrait être au moins trois fois plus rapide pour être exploitable sur les ordinateurs des agences de la banque.

Après une étude qui montre que le parc informatique est vieillissant sur toutes les agences de la banque ; la solution préconisée par la société de services est d'investir dans un matériel trois fois plus rapide. Le gain est important car le matériel devait être rapidement changé en tout état de cause et la solution logicielle retenue sans investissement supplémentaire.

- (a) Est-ce que cet façon de faire est parfois fausse ?
- (b) Est-ce qu'elle est parfois vrai ?

10. Jeu d'échecs

Un jeu d'échecs est un jeu fini. En effet, il y a un nombre de coups déterminés pour aller du début à la fin d'une partie³. Il y a donc un nombre fini de parties possibles.

- (a) Est-ce vrai ?
- (b) Si oui, est-ce qu'un algorithme de complexité linéaire est meilleur qu'un algorithme polynomial ou exponentiel pour faire un partenaire automatique ? Motivez la réponse.

11. Test de primalité : le crible naïf

Le test le plus simple est le suivant : pour tester N , on vérifie s'il est divisible par l'un des entiers compris entre 2 et N (bornes comprises). Si la réponse est négative, alors N est premier, sinon il est composé.

Écrire cet algorithme.

Plusieurs remarques permettent d'améliorer les performances de l'algorithme précédent :

- (a) Il est inutile de chercher un multiple supérieur à la racine carré de k . En effet si $k = pq$, alors soit $p \leq \sqrt{k}$, soit $q \leq \sqrt{k}$
- (b) Il est inutile de chercher si k est divisible par un nombre composé pq si l'on sait déjà qu'il est divisible par p ou par q

12. Test de primalité : le crible d'Ératosthène

Pour savoir les nombres premiers inférieurs à k , nous allons procéder ainsi :

- (a) On forme la liste des entiers de 2 à k

3. La partie est réputée nulle si les joueurs jouent trois fois la même configuration de jeu.

- (b) On retient successivement tous les nombres premiers de la liste. Le premier retenu est 2.
- (c) On barre tous les multiples du nombre retenu à l'étape précédente en commençant par son carré (en effet, $2 \times n$, $3 \times n$, ..., $(n - 1) \times n$ sont déjà barrés, car multiples de 2, 3, ...)
- (d) On s'arrête lorsque l'on cherche un multiple supérieur à la racine carré de k .

Écrire cet algorithme.