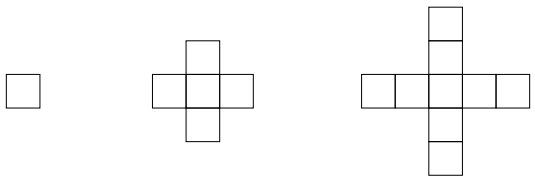
Portail Descartes

Introduction à la science info : Partiel-type

Durée 1h30, documents et calculatrice interdits

amU Faculté des sciences Aix Marseille Université

Exercice 1 (Algorithme informel) Dans la figure suivante sont présentées les trois étapes de construction d'une figure géométrique dont le composant de base est le carré unitaire (c'est-à-dire le carré dont les côtés sont de longueur 1).



1 2 3

1. Soit $n \geq 1$ un entier et soit `figure(n)` la figure obtenue à la fin de la n-ème étape de construction. Expliquez en langage naturel comment obtenir la figure `figure(n+1)` à partir de `figure(n)`.
2. Soit la fonction $c : \mathbb{N} \to \mathbb{N}$ telle que $c(n)$ vaut le nombre de carrés unitaires dessinés dans la figure `figure(n)`. Ainsi, $c(1) = 1$, $c(2) = 5$, $c(3) = 9$. Exprimez $c(n)$ en fonction de $c(n-1)$.
3. En justifiant votre réponse mathématiquement, quelle est la valeur de n pour laquelle la figure associée sera composée de 85 carrés ?

Exercice 2 (Codage des entiers)

1. Donnez le codage en binaire de l'entier naturel 159, en précisant brièvement (5 lignes maximum) votre méthode.
2. Donnez l'entier naturel dont le code binaire est 1001110, en précisant brièvement (5 lignes maximum) votre méthode.
3. Donnez le codage en hexadécimal de l'entier naturel dont le code binaire est 110010011101111 en précisant brièvement (5 lignes maximum) votre méthode.

Exercice 3 (Exécution et analyse d'un algorithme) Considérez l'algorithme suivant, qui prend en entrée un tableau A d'entiers :

```
1  def mystère(A):
2      n = len(A)
3      B = []
4      i = 0
5      x = 0
6      while i < n:
7          B.append(A[i] + x)
8          x = B[i]
9          i = i + 1
10     return B
```

1. Exécutez `mystère([1, 2, 3, 4])` en remplissant la table suivante :

`mystère([1, 2, 3, 4])`

#L	A	n	B	i	x

résultat : ?

2. Combien d'opérations demande l'exécution de `mystère([1, 2, 3, 4])` ?
3. Montrez que l'algorithme `mystère` termine sur n'importe quel tableau d'entiers en entrée.
4. En général, combien d'opérations demande l'exécution de `mystère(A)` pour un tableau A de longueur n ? Calculez d'abord le nombre exact et ensuite simplifiez avec la notation « grand O ».
5. Si B est le résultat du calcul `mystère(A)`, qu'est-ce que la dernière case de B contient ?

Exercice 4 (Écriture d'un algorithme) Écrivez un algorithme `filtre_pairs(A)` qui prend en entrée un tableau A d'entiers et renvoie un tableau qui contient seulement les entiers pairs de A, dans le même ordre. Par exemple, `filtre_pairs([5, 1, 4, 7, 9, 2, 2, 0, 5, -2])` doit renvoyer `[4, 2, 2, 0, -2]`.

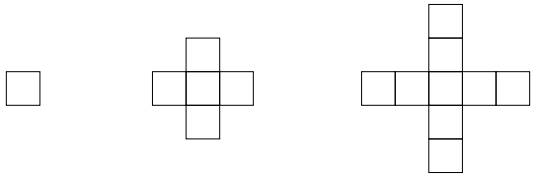
Portail Descartes

Introduction à la science info : Partiel-type

Durée 1h30, documents et calculatrice interdits

amU Faculté des sciences Aix Marseille Université

Exercice 1 (Algorithme informel) Dans la figure suivante sont présentées les trois étapes de construction d'une figure géométrique dont le composant de base est le carré unitaire (c'est-à-dire le carré dont les côtés sont de longueur 1).



1 2 3

1. Soit $n \geq 1$ un entier et soit `figure(n)` la figure obtenue à la fin de la n-ème étape de construction. Expliquez en langage naturel comment obtenir la figure `figure(n+1)` à partir de `figure(n)`.

Solution : Pour obtenir la figure `figure(n+1)`, il faut ajouter quatre carrés unitaires à la figure `figure(n)` de la manière suivante :
- un carré à l'ouest du carré le plus éloigné à gauche du carré central ;
- un carré à l'est du carré le plus éloigné à droite du carré central ;
- un carré au nord du carré le plus éloigné au dessus du carré central ;
- un carré au sud du carré le plus éloigné en dessous du carré central.

2. Soit la fonction $c : \mathbb{N} \to \mathbb{N}$ telle que $c(n)$ vaut le nombre de carrés unitaires dessinés dans la figure `figure(n)`. Ainsi, $c(1) = 1$, $c(2) = 5$, $c(3) = 9$. Exprimez $c(n)$ en fonction de $c(n-1)$.

Solution : À chaque étape, on ajoute 4 carrés à la figure de l'étape précédente, d'où : $c(n) = c(n-1) + 4$.

3. En justifiant votre réponse mathématiquement, quelle est la valeur de n pour laquelle la figure associée sera composée de 85 carrés ?

Solution : L'idée est maintenant de définir la fonction c non pas de manière récursive mais en fonction de n, directement. Étant donné que l'on ajoute 4 carrés à chaque étape, la quantité de carrés est linéaire en fonction du nombre d'étapes. Plus précisément, le nombre de carrés est défini par $c(n) = 4n + k$. Il reste à déterminer la valeur de k. Pour ce faire, prenons une valeur de n quelconque. Avec $n = 1$, on a $c(n) = 1 = 4 \times 1 + k$. En résolvant cette équation du premier degré, on obtient $k = -3$. Par conséquent :

$$c(n) = 4n - 3.$$

Ainsi, pour obtenir une figure composée de 85 carrés, il faut trouver n tel que $c(n) = 85$, c'est-à-dire résoudre l'équation :

$$4n - 3 = 85 \iff 4n = 88 \iff n = \frac{88}{4} = 22.$$

Exercice 2 (Codage des entiers)

1. Donnez le codage en binaire de l'entier naturel 159, en précisant brièvement (5 lignes maximum) votre méthode.

Solution : On commence par retrouver les puissances de 2 successives :

n	0	1	2	3	4	5	6	7	8
2^n	1	2	4	8	16	32	64	128	256

Dans 159, on peut donc placer $128 = 2^7$ comme plus grande puissance de 2. Il reste $159 - 128 = 31$ qui s'écrit $16 + 8 + 4 + 2 + 1$ (en effet, $31 = 32 - 1$ est un entier précédent une puissance de 2). Ainsi, l'écriture en binaire de l'entier naturel 159 est 10011111.

2. Donnez l'entier naturel dont le code binaire est 1001110, en précisant brièvement (5 lignes maximum) votre méthode.

Solution : On utilise la table de la question précédente pour sommer les puissances de 2 correspondantes :

$$2^1 + 2^2 + 2^3 + 2^6 = 2 + 4 + 8 + 64 = 78$$

3. Donnez le codage en hexadécimal de l'entier naturel dont le code binaire est 110010011101111 en précisant brièvement (5 lignes maximum) votre méthode.

Solution : On groupe les chiffres binaires par 4 à partir de la droite, en ajoutant des zéros en tête si nécessaire, et pour chaque groupe on prend le chiffre hexadécimal correspondant :

$$\underbrace{0110}_{6}\underbrace{0100}_{4}\underbrace{1110}_{E}\underbrace{1111}_{F}$$

Ce qui nous donne le résultat de 64EF.

Exercice 3 (Exécution et analyse d'un algorithme) Considérez l'algorithme suivant, qui prend en entrée un tableau A d'entiers :

```
1  def mystère(A):
2      n = len(A)
3      B = []
4      i = 0
5      x = 0
6      while i < n:
7          B.append(A[i] + x)
8          x = B[i]
9          i = i + 1
10     return B
```

1. Exécutez `mystère([1, 2, 3, 4])` en remplissant la table suivante :

`mystère([1, 2, 3, 4])`

#L	A	n	B	i	x

résultat : ?

Solution :

`mystère([1, 2, 3, 4])`

#L	A	n	B	i	x
1	[1, 2, 3, 4]				
2		4			
3			[]		
4				0	
5					0
6					
7			[1]		
8					1
9				1	
6					
7			[1, 3]		
8					3
9				2	
6					
7			[1, 3, 6]		
8					6
9				3	
6					
7			[1, 3, 6, 10]		
8					10
9				4	
6					
10					

résultat : [1, 3, 6, 10]

2. Combien d'opérations demande l'exécution de `mystère([1, 2, 3, 4])` ?

Solution : 23 opérations, le nombre de lignes de la table dans la question précédente.

3. Montrez que l'algorithme `mystère` termine sur n'importe quel tableau d'entiers en entrée.

Solution : La seule raison pour laquelle l'algorithme ne pourrait pas terminer est la boucle `while`. La condition de la boucle est $i < n$, où n est la longueur du tableau (un entier ≥ 0) et i a initialement la valeur 0. Si $n = 0$ alors on n'entre jamais dans la boucle `while`, mais en tout cas, puisque i est incrémentée à la ligne 9 à chaque tour de boucle, alors que n ne change pas, tôt ou tard i va atteindre la valeur n, ce qui rend la condition de la ligne 6 fausse et cause la terminaison de la boucle. L'algorithme termine donc à la ligne suivante avec l'instruction `return`.

4. En général, combien d'opérations demande l'exécution de `mystère(A)` pour un tableau A de longueur n ? Calculez d'abord le nombre exact et ensuite simplifiez avec la notation « grand O ».

Solution : Les lignes 1, 2, 3, 4, 5 et 10 sont exécutées exactement une fois, puisque elles sont en dehors de la boucle. L'intérieur de la boucle (lignes 7, 8, 9) est exécuté une fois pour chaque valeur $i = 0, 1, \ldots, n-1$, donc n fois en total. La ligne 6, la tête de la boucle, est exécutée une fois de plus (il y a un dernier test pour vérifier que la condition est devenue fausse), donc $n + 1$ fois.
En total cela donne $6 + 3n + n + 1 = 4n + 7$ opérations, soit $O(n)$.

5. Si B est le résultat du calcul `mystère(A)`, qu'est-ce que la dernière case de B contient ?

Solution : La somme des valeurs de A. En fait, la case i de B contient la somme $A[0] + \cdots + A[i]$ pour tout $i = 0, \ldots, n-1$.

Exercice 4 (Écriture d'un algorithme) Écrivez un algorithme `filtre_pairs(A)` qui prend en entrée un tableau A d'entiers et renvoie un tableau qui contient seulement les entiers pairs de A, dans le même ordre. Par exemple, `filtre_pairs([5, 1, 4, 7, 9, 2, 2, 0, 5, -2])` doit renvoyer `[4, 2, 2, 0, -2]`.

Solution :

```
def filtre_pairs(A):
    n = len(A)
    B = []
    for i in range(n):
        if A[i] % 2 == 0:
            B.append(A[i])
    return B
```