DS 4 d'informatique de l'année 2022/2023 - MPSI

Calculatrices interdites. Pensez à numéroter vos feuilles.
Sur votre copie, les questions doivent apparaître dans l'ordre du sujet.
Si vous repérez une erreur d'énoncé, signalez le sur votre copie et poursuivez votre composition.
Le sujet est composé de trois exercices indépendants.

Exercice 1. Représentation des nombres

Les détails de vos calculs doivent apparaître sur votre copie.

1. (a) Quelle est l'écriture en base 2 de 88 ?
   (b) Quelle est l'écriture en base 10 de $(1001\ 0110)_2$?
2. Sans convertir en base 10, calculer : $(1101\ 1101)_2+(101\ 0101)_2$, $(1001\ 0110)_2-(111\ 1001)_2$ et $(11\ 1011)_2\times(1101)_2$. Donner les résultats en base 2.
3. (a) Sans convertir en base 10, calculer le quotient et le reste dans la division euclidienne de $(1111\ 1010)_2$ par $(101)_2$. Donner le résultat en base 2.
   (b) Vérifier les valeurs obtenues dans la question 3a en utilisant la base 10.
4. (a) Quelle est la représentation complément à deux de -143 ?
   (b) Quel est le nombre dont la représentation complément à deux est $(\underbrace{1\dots1}_{54 \text{ uns}}01\ 1010\ 1001)$?

Exercice 2. Représentation de Zeckendorf

Dans cet exercice, on s'intéresse à la célèbre suite de Fibonacci qu'on utilisera pour encoder des nombres, puis pour implémenter une exponentiation rapide.

Suite de Fibonacci. La suite de Fibonacci $(\mathcal{F}_k)_{k\in\mathbb{N}}$ est définie par :

$$\begin{cases} \mathcal{F}_0 = 1, \mathcal{F}_1 = 2 \\ \mathcal{F}_{k+2} = \mathcal{F}_{k+1} + \mathcal{F}_k \qquad \text{pour tout } k \geqslant 0 \end{cases}$$

1. Donner les dix premiers termes de la suite de Fibonacci.
2. (a) Écrire une fonction `fibo` qui prend en entrée un entier $\mathtt{k} \in \mathbb{Z}$ et renvoie la liste `[`$\mathcal{F}_0$`, `$\mathcal{F}_1$`, ..., `$\mathcal{F}_\mathtt{k}$`]`. Si `k < 0`, votre fonction déclenchera une erreur.
   (b) Montrer la terminaison de la fonction `fibo`.
   (c) Montrer la correction de la fonction `fibo`.
   (d) Donner en les justifiant les complexités spatiales et temporelles de la fonction `fibo`.

Représentation de Zeckendorf. Dans la suite, on dira qu'une somme d'entiers est ***valide*** si les quatre conditions suivantes sont respectées :

- Chaque terme de la somme est un élément de la suite de Fibonacci.
- Ces termes sont strictement décroissants.
- Si on note $a \in \mathbb{N}^*$ la valeur de la somme et k l'entier tel que $\mathcal{F}_k \leqslant a < \mathcal{F}_{k+1}$, alors $\mathcal{F}_k$ apparaît dans la somme.
- Pour tout $k \in \mathbb{N}$, si $\mathcal{F}_k$ apparaît dans la somme alors $\mathcal{F}_{k+1}$ n'y apparaît pas.

On admettra que tout entier strictement positif $a \in \mathbb{N}^*$ peut s'écrire comme une unique somme valide. Par exemple, avec $a = 40$, on obtient $a = 40 = 34 + 5 + 1 = \mathcal{F}_7 + \mathcal{F}_3 + \mathcal{F}_0$.

3. Écrire $a = 60$ comme une somme valide.

À chaque entier $a \in \mathbb{N}^*$ on associe une chaîne de caractères `s` contenant une suite de `0` et de `1` appelée ***représentation de Zeckendorf***. Pour obtenir `s`, on écrit a comme une somme valide, puis on définit `s` comme étant la chaîne de caractères telle que :

– Si $\mathcal{F}_\mathrm{k}$ apparaît dans la somme alors `s[k]` vaut `"1"`.
– Sinon, `s[k]` vaut `"0"`.

Afin de garantir l'unicité, on impose que le dernier caractère de `s` soit un `1`. Par exemple, la représentation de Zeckendorf de $a = 40$ est `"10010001"`.

4. Quelle est la représentation de Zeckendorf de $a = 60$?
5. Écrire une fonction `zeckToInt` qui prend en entrée une chaîne de caractères `s` et renvoie l'entier a dont `s` est la représentation de Zeckendorf.
6. (a) Écrire une fonction `tailleZeck` qui prend en entrée un entier a et renvoie la taille de la représentation de Zeckendorf de a notée ℓ. On garantira une complexité en $\Theta(\ell)$.
   (b) Écrire une fonction `intToZeck` qui prend en entrée un entier et renvoie sa représentation de Zeckendorf.

Code de Fibonacci. On souhaite maintenant encoder une suite de nombres $a_0, a_1, \dots a_{n-1} \in \mathbb{N}^*$ par une suite de `0` et de `1` appelée ***code de Fibonacci***. Pour cela, on écrit la représentation de Zeckendorf de a_0, puis on y concatène un `1`, puis on y concatène la représentation de Zeckendorf de a_1, puis on y concatène un `1`, ..., puis on y concatène la représentation de Zeckendorf de a_{n-1} et enfin on y concatène un `1`. Par exemple, le code de Fibonacci de la suite $5, 40, 21$ est `0001110010001100000011`.

En Python, la suite de nombres $a_0, a_1, \dots a_{n-1}$ sera représentée par une liste « `A: list[int]` » de taille n et le code de Fibonacci correspondant sera représenté par une chaîne de caractères `CF`.

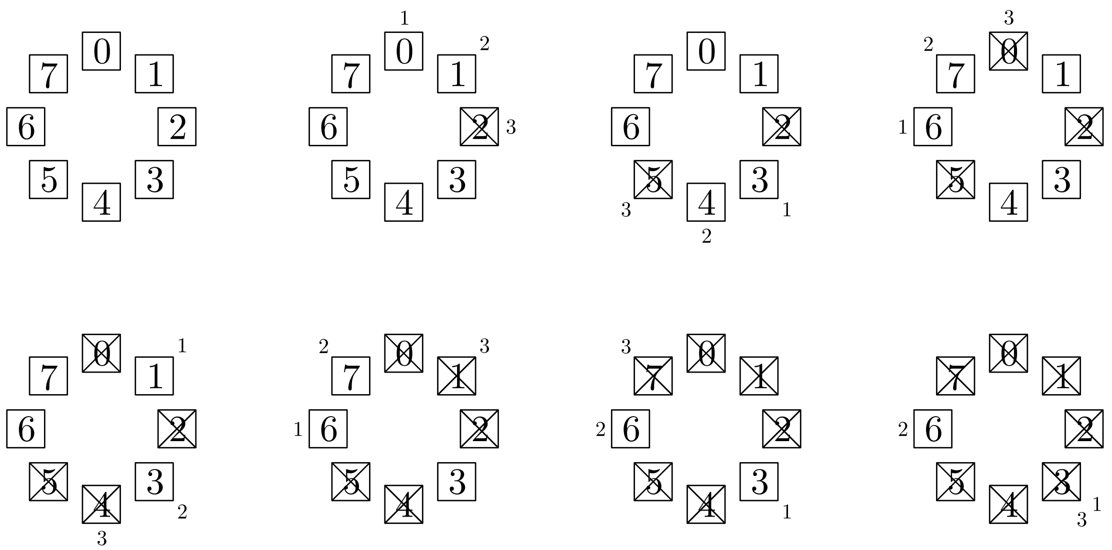
7. (a) Écrire une fonction `AtoCF` qui prend en entrée `A` et renvoie `CF`.
   (b) Écrire une fonction `CFtoA` qui prend en entrée `CF` et renvoie `A`. On expliquera succinctement la procédure utilisée.

Exponentiation rapide. Le but de cette partie est d'utiliser la représentation de Zeckendorf d'un entier $a \in \mathbb{N}^*$ pour calculer rapidement x^a où $x \in \mathbb{R}$.

8. (a) Sans utiliser l'opérateur `**`, écrire une fonction `puiss` qui prend en entrée une chaîne de caractères `s` contenant la représentation de Zeckendorf d'un entier $a \in \mathbb{N}^*$, ainsi qu'un flottant $x \in \mathbb{R}$, et renvoie x^a. Votre fonction devra manipuler directement `s`, sans jamais calculer explicitement la valeur de a. On expliquera succinctement la procédure utilisée.
   (b) Donner en le justifiant le temps d'exécution de la fonction `puiss` en fonction de a.

Exercice 3. Plouf-plouf

Dans une cour d'école, $n \in \mathbb{N}^*$ enfants souhaitent choisir le capitaine de l'équipe de football en utilisant la technique du « plouf-plouf ». Pour cela, ils se numérotent de 0 à $n-1$, se placent en cercle en suivant la numérotation et choisissent un entier naturel non nul $m \in \mathbb{N}^*$ (en pratique, cet entier est le nombre de syllabes dans une comptine du genre « Une boule en or c'est toi qui sors »). L'un des enfants parcourt ses camarades, élimine la $m^{\text{ème}}$ personne rencontrée ; puis la personne située m positions plus loin parmi les enfants restants ; puis la personne située m positions plus loin parmi les enfants restants ; et ainsi de suite jusqu'à ce qu'il ne reste plus qu'un seul enfant. Par exemple, pour $n = 8$ et $m = 3$, l'ordre des éliminations est le suivant : $2, 5, 0, 4, 1, 7, 3$. Le capitaine de l'équipe sera donc le numéro 6.



1. Écrire une fonction `capitaine` qui prend en entrée les entiers n, m et renvoie le numéro de l'enfant qui sera choisi pour être capitaine de l'équipe.
2. (a) On se place dans le cas particulier où $m = 2$ et $n = 2^k$ pour un certain $k \in \mathbb{N}$. Quel sera le numéro du capitaine ? Le montrer.
   (b) On se place dans le cas particulier où $m = 2$. En utilisant l'écriture en base 2 de n, proposer une manière de déterminer le numéro du capitaine.
   (c) En déduire une fonction `capitaine2` qui prend en entrée un entier $n \in \mathbb{N}^*$ et renvoie le numéro du capitaine dans le cas où $m = 2$. Le temps d'exécution de votre fonction devra être en $\mathcal{O}(\log(n))$.

Remarque. Ce problème s'appelle le ***problème de Josèphe*** et fut énoncé pour la première fois par Flavius Josèphe au 1^{er} siècle après JC (dans le cas de soldats romains sur le point d'exécuter des prisonniers).