Points fixes de fonctions à domaine fini (d'après X mp 2013)

Durée : 2 heures

Dans ce problème on s'intéresse aux points fixes des fonctions $f : \mathrm{E} \to \mathrm{E}$, où E est un ensemble fini. Le calcul effectif et efficace des points fixes de telles fonctions est un problème récurrent en informatique (transformation d'automates, vérification automatique de programmes, algorithmique des graphes, etc) et admet différentes approches selon la structure de E et les propriétés de f.

On suppose par la suite un entier $n > 0$ fixé et on pose $\mathrm{E}_n = \{0, 1, \ldots, n-1\}$. On représente une fonction $f : \mathrm{E}_n \to \mathrm{E}_n$ par un tableau `t` de taille n, autrement dit $f(x) = $ `t[x]` pour tout $x = 0, 1, \ldots, n-1$. Ainsi, la fonction f_0 qui à $x \in \mathrm{E}_{10}$ associe $2x+1$ modulo 10 est-elle représentée par le tableau :

t[0]	t[1]	t[2]	t[3]	t[4]	t[5]	t[6]	t[7]	t[8]	t[9]
1	3	5	7	9	1	3	5	7	9

Toutes les fonctions demandées devront être écrites en Python.

Partie I. Recherche de point fixe, cas général

On rappelle que x est un *point fixe* de la fonction f si et seulement si $f(x) = x$.

Question 1. Écrire une fonction `admet_point_fixe(t)` qui prend en argument un tableau `t` de taille n et renvoie `True` si la fonction $f : \mathrm{E}_n \to \mathrm{E}_n$ représentée par `t` admet un point fixe, `False` sinon. Par exemple, `admet_point_fixe` devra renvoyer `True` pour le tableau donné en introduction, puisque 9 est un point fixe de la fonction f_0 qui à x associe $2x+1$ modulo 10.

Question 2. Écrire une fonction `nb_points_fixes(t)` qui prend en argument un tableau `t` de taille n et renvoie le nombre de points fixes de la fonction $f : \mathrm{E}_n \to \mathrm{E}_n$ représentée par `t`. Par exemple, `nb_points_fixes` devra renvoyer 1 pour le tableau donné en introduction, puisque 9 est le seul point fixe de f_0.

On note f^k l'itérée k-ième de f, autrement dit $f^k = \underbrace{f \circ f \circ \cdots \circ f}_{k \text{ fois}}$.

Question 3. Écrire une fonction `itere(t, x, k)` qui prend en premier argument un tableau `t` de taille n représentant une fonction $f : \mathrm{E}_n \to \mathrm{E}_n$, en deuxième et troisième arguments des entiers x et k de E_n, et renvoie $f^k(x)$.

Question 4. Écrire une fonction `nb_points_fixes_iteres(t, k)` qui prend en premier argument un tableau `t` de taille n représentant une fonction $f : \mathrm{E}_n \to \mathrm{E}_n$, en deuxième argument un entier $k \geqslant 0$, et renvoie le nombre de points fixes de f^k.

Un élément $z \in \mathrm{E}_n$ est dit *attracteur principal* de $f : \mathrm{E}_n \to \mathrm{E}_n$ si et seulement si z est un point fixe de f, et pour tout $x \in \mathrm{E}_n$, il existe un entier $k \geqslant 0$ tel que $f^k(x) = z$.

Afin d'illustrer cette notion, on pourra vérifier que la fonction f_1 représentée par le tableau ci-dessous admet 2 comme attracteur principal.

t[0]	t[1]	t[2]	t[3]	t[4]	t[5]	t[6]
5	5	2	2	0	2	2

En revanche, on notera que la fonction f_0 donnée en introduction n'admet pas d'attracteur principal, puisque $f_0^k(0) \neq 9$ quel que soit l'entier $k \geqslant 0$.

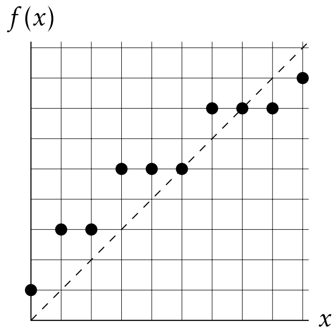
Question 5. Écrire une fonction `admet_attracteur_principal(t)` qui prend en argument un tableau `t` de taille n et renvoie `True` si et seulement si la fonction $f : \mathrm{E}_n \to \mathrm{E}_n$ représentée par `t` admet un attracteur principal, `False` sinon. On ne requiert pas ici une solution efficace.

On suppose à la question 6 que f admet un attracteur principal. Le *temps de convergence* de f en $x \in \mathrm{E}_n$ est le plus petit entier $k \geqslant 0$ tel que $f^k(x)$ soit un point fixe de f. Pour la fonction f_1 ci-dessus, le temps de convergence en 4 est égal à 3. En effet, $f_1(4) = 0$, $f_1^2(4) = 5$, $f_1^3(4) = 2$ et 2 est un point fixe de f_1. On note $\mathrm{tc}(f, x)$ le temps de convergence de f en x.

Question 6. Écrire une fonction `temps_de_convergence(t, x)` qui prend en argument un tableau `t` de taille n représentant une fonction $f : \mathrm{E}_n \to \mathrm{E}_n$ qui admet un attracteur principal, en deuxième argument un entier x de E_n, et renvoie le temps de convergence de f en x.

Partie II. Recherche efficace de points fixes

Toute fonction `point_fixe(t)` retournant un point fixe d'une fonction arbitraire est de complexité au mieux linéaire en n.
On s'intéresse maintenant à des améliorations possibles de cette complexité lorsque le fonction considérée est croissante.
On rappelle qu'une fonction $f : \mathrm{E}_n \to \mathrm{E}_n$ est croissante si et seulement si pour tout $(x, y) \in \mathrm{E}_n^2$, $x \leqslant y \Rightarrow f(x) \leqslant f(y)$.
On admet qu'une fonction croissante de E_n dans E_n admet toujours un point fixe.
À titre d'exemple, la fonction dont le tableau et le graphe sont donnés ci-dessous est croissante. Elle a deux points fixes, à savoir les entiers 5 et 7.



t[0]	t[1]	t[2]	t[3]	t[4]	t[5]	t[6]	t[7]	t[8]	t[9]
1	3	3	5	5	5	7	7	7	8

Question 7. Ecrire une fonction `est_croissante(t)` qui prend en argument un tableau `t` et renvoie `True` si la fonction représentée par `t` est croissante, `False` sinon. On impose un temps de calcul linéaire en la taille n du tableau. On ne demande pas de démonstration du fait que le temps de calcul de la solution proposée est linéaire.

Question 8. Ecrire une fonction `point_fixe(t)` qui prend en argument un tableau `t` de taille n représentant une fonction croissante $f : \mathrm{E}_n \to \mathrm{E}_n$ et retourne un entier $x \in \mathrm{E}_n$ tel que $f(x) = x$. On impose un temps de calcul logarithmique en la taille n du tableau. On ne demande pas de démonstration du fait que le temps de calcul de la solution proposée est logarithmique.

♦ ♦
♦