

Double hélice à distance constante

On s'attachera plus à écrire du code correct qu'à l'efficacité du code. Avant d'implémenter une fonction, on donnera toujours le type attendu.

On se donne un type `type point = int * int;`. Dans cet exercice, on n'utilisera pas les fonctions du module List fourni par OCaml.

1. Donnez une implémentation possible du type `type figure = point list;`.
2. Implémentez une fonction qui, étant donnée une figure, renvoie la liste des abscisses des points de la figure.
3. Implémentez une fonction qui, étant données une liste d'abscisses et une liste d'ordonnées, renvoie la figure des points ayant les abscisses et les ordonnées correspondantes.
4. Implémentez une fonction qui étant données deux figures, renvoie la figure (sans doublon) des points des deux figures.
5. Implémentez une fonction qui, étant données une figure, et un prédicat (fonction renvoyant vrai ou faux) prenant en entrée un point, renvoie une liste de figures selon la valeur du prédicat.
6. Expliquez comment généraliser la fonction de la question précédente à une fonction à valeurs quelconques.
7. Bonus : Donnez une possible implémentation plus efficace pour la question 4.
8. Très Bonus (vraiment très difficile, en fait) : Donnez une implémentation plus efficace pour la question 5.

Polynomialité

On s'attachera plus à écrire du code correct qu'à l'efficacité du code. Avant d'implémenter une fonction, on donnera toujours le type attendu.

On s'intéresse aux équations de la forme $c_3x^3 + c_2x^2 + c_1x + c_0 = 0$. On les représentera en OCaml par le type `type eq = float list;`. On appelle degré d'une telle équation la valeur maximale de i tel que $c_i x^i \neq 0$. On utilisera le type OCaml suivant pour représenter des ensembles de solutions `type solution = | None | Some of float list | R;`.

1. Implémentez une fonction qui a une équation renvoie son degré.
2. Implémentez une fonction qui, a une équation de degré exactement 0 renvoie l'ensemble, potentiellement vide, de ses solutions.
3. Implémentez une fonction qui, a une équation de degré exactement 1 renvoie l'ensemble, potentiellement vide, de ses solutions.
4. Implémentez une fonction qui, a deux ensembles de solutions, vérifie si ils sont compatibles.
5. Implémentez une fonction qui, a une équation de degré exactement 2 renvoie l'ensemble, potentiellement vide, de ses solutions réelles.
6. Implémentez une fonction qui, étant donné une équation, renvoie son ensemble de solution. On renverra pour l'instant None, dans le cas ou la fonction solution n'est pas implémentée.
7. Implémentez une fonction qui, a une équation de degré exactement 3 renvoie l'ensemble, potentiellement vide, de ses solutions réelles. On pourra utiliser, sans démonstration, la formule suivante :

$$x = \sqrt[3]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)} + \sqrt[2]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3} + \sqrt[3]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)} - \sqrt[2]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3} - \dots$$

On s'attachera plus à écrire du code correct qu'à l'efficacité du code. Avant d'implémenter une fonction, on donnera toujours le type attendu.

On rappelle qu'une permutation sur X est une bijection de X dans X .

1. Donnez le nombre de permutations de $\llbracket 1, n \rrbracket$ en fonction de n .
2. Montrez que $\sigma : X \rightarrow X$ est injective si et seulement si elle est bijective.
3. Implémentez une fonction qui, étant donnés une fonction et un entier n , vérifie si c'est une permutation de $\llbracket 1, n \rrbracket$.
4. Implémentez une fonction qui, étant donnée une 'a list de longueur n et une permutation de $\llbracket 1, n \rrbracket$, renvoie la liste ayant eu ses éléments permutés.

L'ordre d'une permutation est le nombre de fois minimal qu'il faut l'appliquer avant de retomber sur l'identité.

5. Implémentez une fonction qui étant donnés une permutation et un nombre k renvoie sa k -ième itérée.
6. Implémentez une fonction qui étant donné une permutation renvoie son ordre.
7. Quel est l'ordre maximal d'une permutation ?