

TP ALBA algorithme d'évaluation et d'interpolation rapide

13 Novembre 2020

On accordera encore une fois beaucoup d'importance à respecter les signatures des fonctions.

Il est attendu un unique fichier `tp4_nom_prenom.sage` qui doit pouvoir être loadé en terminal pour pouvoir être tester. Le fichier ne doit comporter rien de plus que les fonctions qui sont demandées dans le tp et les fonctions auxiliaires. En particulier, aucune variable ne doit être instanciée en dehors des fonctions et le fichier ne doit exécuter aucun code. Je vous invite à commenter votre code pour aider à sa clarté.

Exercice 1 :

On veut écrire l'algorithme de calcul de l'arbre des sous-produits pour des points a_0, \dots, a_n donnés.

1. Écrire d'abord la classe `Tree` avec le code suivant :

```
class Tree(object):
    def __init__(self):
        self.right = None
        self.left = None
        self.root = None
```

On pourra par la suite si on le souhaite améliorer le constructeur en prenant par exemple en argument la racine directement.

2. Écrire l'algorithme rapide de calcul de l'arbre des sous-produits. La fonction s'appellera `product_tree` et prendra en entrée la liste des éléments a_0, \dots, a_n et l'anneau de polynômes auquel doit appartenir les polynômes des sous-produits et doit renvoyer un élément de la classe `Tree` dont chaque noeud est un des sous-produits.
3. Écrire la fonction `fast_eval` qui prend en entrée un polynôme et une liste de points et qui renvoie la liste des valeurs du polynôme en ces points.

Exercice 2 :

On s'occupe maintenant de l'interpolation.

1. Implémenter la fonction `sum_fractions` qui prend en entrée trois arguments : deux listes de coefficients de même longueur la première contenant les c_i , la deuxième contenant les a_i et l'anneau de polynômes auxquels on veut que le résultat appartienne et qui renvoie le numérateur et le dénominateur de la fraction rationnelle $\frac{N}{P} = \sum_i \frac{c_i}{X-a_i}$ avec $P = (X - a_0) \cdots (X - a_n)$.
2. Écrire la fonction `fast_interpolation` qui prend entrée deux listes de longueur $n + 1$ et un anneau de polynôme auquel on veut que le résultat appartienne. La première liste contient les a_i , la deuxième contient les b_i et la fonction retourne l'unique polynôme P de degré au plus n tel que $P(a_i) = b_i$.