

# INFO0054 - Programmation fonctionnelle

## Répétition 8

Jean-Michel BEGON

21 avril 2020

### Inclusion/exclusion

---

#### Exercice 1.

Écrire une fonction `nbsum` qui prend comme argument un nombre  $n$  et qui renvoie le nombre de façons d'écrire une somme égale à  $n$  (on comptera une seule fois les commutations).

---

#### Exercice 2.

Écrire la fonction `lagrange`, prenant comme argument un entier naturel  $n$  et renvoyant la liste de tous les quadruplets d'entiers naturels  $(a, b, c, d)$  tels que  $a^2 + b^2 + c^2 + d^2 = n$ .

```
(lagrange 13) ==>
((0 0 2 3) (0 0 3 2) (0 2 0 3) (0 2 3 0) (0 3 0 2) (0 3 2 0)
 (1 2 2 2) (2 0 0 3) (2 0 3 0) (2 1 2 2) (2 2 1 2) (2 2 2 1)
 (2 3 0 0) (3 0 0 2) (3 0 2 0) (3 2 0 0))
```

```
(lagrange 18)
((0 0 3 3) (0 1 1 4) (0 1 4 1) (0 3 0 3) (0 3 3 0) (0 4 1 1)
 (1 0 1 4) (1 0 4 1) (1 1 0 4) (1 1 4 0) (1 2 2 3) (1 2 3 2)
 (1 3 2 2) (1 4 0 1) (1 4 1 0) (2 1 2 3) (2 1 3 2) (2 2 1 3)
 (2 2 3 1) (2 3 1 2) (2 3 2 1) (3 0 0 3) (3 0 3 0) (3 1 2 2)
 (3 2 1 2) (3 2 2 1) (3 3 0 0) (4 0 1 1) (4 1 0 1) (4 1 1 0))
```

---

#### Exercice 3.

Une *tricoupure* d'une liste  $\ell$  est une liste de trois listes non vides dont la concaténation (dans l'ordre) vaut  $\ell$ . Écrire une fonction `3-cuts` qui à toute liste  $\ell$  associe la liste des tricoupures de  $\ell$ .

Par exemple, si  $\ell$  est `(a b)` la liste des tricoupures de  $\ell$  est la liste vide; si  $\ell$  est `(a b c d)` la liste des tricoupures de  $\ell$  comporte, dans un ordre quelconque, les trois listes `((a b) (c) (d))`, `((a) (b c) (d))` et `((a) (b) (c d))`.

#### **Variante**

Une *tricoupure* d'une liste  $\ell$  est une liste de trois listes dont la concaténation (dans l'ordre) vaut  $\ell$ . Écrire une fonction `tricoup-lv` qui à toute liste  $\ell$  associe la liste des tricoupures de  $\ell$ .

Par exemple, si  $\ell$  est `(a)` la liste des tricoupures de  $\ell$  comporte, dans un ordre quelconque, les trois listes `((a) () ())`, `(() (a) ())` et `(() () (a))`.

# Arbres

---

## **Exercice 4.**

Générer à partir d'une liste, tous les arbres binaires complets dont la lecture des feuilles de gauche à droite donne la liste.

Par exemple, si on représente un nœud par une liste et une feuille par l'atome étiquette.

```
(binary-trees '(a b c d)) ==>  
((a (b (c d))) ((a b) (c d)) (a ((b c) d)) ((a (b c)) d)  
 ((a b) c) d))
```