

# INFO0054 - Programmation fonctionnelle

## Répétition 8: Graphes, inclusion/exclusion et récursivité forte de nombres

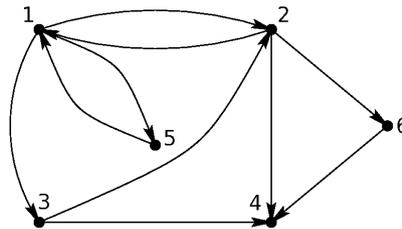
Jean-Michel BEGON

24 Avril 2018

### Graphes

Nous convenons de représenter un *graphe orienté* comportant  $n$  nœuds par une liste de  $n$  listes ; la liste correspondant au nœud  $x$  a pour premier élément  $x$  et pour éléments suivants les successeurs (immédiats) de  $x$ .

Par exemple le graphe



sera représenté par la liste

```
'((1 2 3 5) (2 1 4 6) (3 2 4) (4) (5 1) (6 4))
```

---

#### Exercice 1.

Écrire une fonction `adj` qui prend comme argument la représentation d'un graphe orienté `g`, ainsi qu'un nœud `n` de `g`, et qui retourne la liste des nœuds adjacents de `n`.

---

#### Exercice 2.

Écrire le prédicat `(path? g n1 n2)` prenant deux nœuds `n1` et `n2` et la représentation `g` d'un graphe orienté en arguments et retournant `#t` s'il existe un chemin de `n1` vers `n2` dans `g` et `#f` sinon.

---

#### Exercice 3.

Écrire une fonction `inv` qui, à partir d'un graphe orienté, construit le graphe retourné correspondant.

## Inclusion/exclusion

---

### Exercice 4.

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 5.

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))
```

## Récurtivité forte de nombres

---

### Exercice 6.

Écrire une fonction *efficace* permettant de calculer

$$f(n) = \sum_{i=0}^{n-1} f(n-i-1)^{f(i)}$$

---

### Exercice 7.

Écrire une fonction *efficace* permettant de calculer

$$f(n) = \sum_{i=0}^{n-1} (((f(i) + 2) \times (f(n-i-1) + 3)) \bmod (n^2 + i + 5))$$