TD 1

1 Quicksort

Écrire l'algorithme qui trie un tableau $a_1 a_2 \dots a_n$ par la méthode *quicksort*, décrite ci-dessous. Soit $a_\ell a_{\ell+1} \dots a_r$ un sous-tableau ($1 \leq \ell < r \leq n$) et on note $v = a_r$.

- On range les éléments $a_\ell, a_{\ell+1}, \dots, a_r$ de telle manière que
  - $a_i \leq v$ pour $\ell \leq i < k$
  - $a_k = v$
  - $a_i > v$ pour $k < i \leq r$
- a_k est bien placé dans le tableau et on recommence, si besoin, avec les sous-tableaux $a_\ell \dots a_{k-1}$ et $a_{k+1} \dots, a_r$

2 Suite de Lucas

La suite de Lucas est définie par :

$$\ell_n = \begin{cases} 2 & \text{si} \quad n = 0, \\ 1 & \text{si} \quad n = 1, \\ \ell_{n-1} + \ell_{n-2} & \text{si} \quad n > 1. \end{cases}$$

1. Écrire un algorithme permettant de calculer les 20 premiers nombres de Lucas;
2. Prouver que $\ell_n = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n$.

3 Nombre de Stirling

Le nombre de possibilités pour partitionner un ensemble de n éléments en k sous-ensembles non-vides est le nombre de Stirling de deuxième type, et est noté $s_{n,k}$. Par exemple, l'ensemble $\{1, 2, 3\}$ peut être partitionné

- en trois sous-ensembles d'une seule façon : $\{1\}, \{2\}, \{3\}$
- en deux sous-ensembles de trois façons :
  - $\{1, 2\}, \{3\}$
  - $\{1, 3\}, \{2\}$
  - $\{1\}, \{2, 3\}$

- en un seul sous-ensemble d'une seule façon : $\{1, 2, 3\}$,

donc, $s_{3,1} = 1$, $s_{3,2} = 3$, $s_{3,3} = 1$. Pour $s_{n,k}$, $1 \leq k \leq n$ on a la relation récursive suivante :

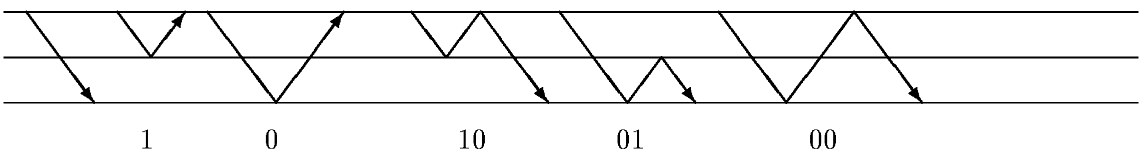
$$s_{n,k} = \begin{cases} 1 & \text{si} \quad k = 1 \text{ ou } k = n \\ s_{n-1,k-1} + k s_{n-1,k} & \text{si} \quad 1 < k < n. \end{cases}$$

Écrire une fonction récursive qui calcule $s_{n,k}$.

4 Séquences de Fibonacci

1. Supposons que nous mettions deux plaques de verre face contre face. De combien de façons possibles a_n un rayon de lumière peut-il traverser les deux plaques ou être réfléchi aprés avoir changé de direction n fois.

$a_0 = 1 \quad a_1 = 2 \qquad\qquad a_2 = 3$



2. Soit F_n la liste de toutes les chaînes binaires qui ne contiennent pas deux *1* qui se suivent; par exemple F_1, F_2, F_3 et F_4 sont :
$F_1 = \{0, 1\}$
$F_2 = \{00, 01, 10\}$
$F_3 = \{000, 001, 010, 100, 101\}$
$F_4 = \{0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010\}$
F_n est définie récursivement par la relation suivante :

$$F_n = \begin{cases} \emptyset & \text{si} \quad n = 0, \\ \{0, 1\} & \text{si} \quad n = 1, \\ \mathbf{0}F_{n-1} \cup \mathbf{10}F_{n-2} & \text{si} \quad n > 1. \end{cases} \tag{1}$$

En d'autres mots F_0 est la liste vide, F_1 est la liste formée de deux chaînes de longueur 1 : *0* et *1* respectivement, et généralement, F_n est obtenue par l'union de la liste F_{n-1} après que l'on a ajouté *0* à chaque chaîne et de la liste F_{n-2} après que l'on a ajouté *10* à chaque chaîne.

(a) Les deux ensembles définis au point 1. et 2. sont en *bijection*
(b) Écrire la définition de la fonction *generer* qui prend un entier n en argument et retourne la liste F_n.
NB L'ensemble F_n est compté avec le nombre de Fibonacci.