L'idée du TP est de comparer l'efficacité de différentes méthodes de calcul de termes d'une même suite expérimentalement. On utilise les commandes `%time` *ou* `cputime()`*. On peut comparer des nombres, représenter des courbes...*

Exercice 1

Comparer expérimentalement les deux méthodes de calcul de pgcd (Euclide et binaire).

Exercice 2

Comparer plusieurs méthodes pour calculer les coefficients binomiaux :

1. L'expression : $\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$
2. La formule de récurrence donnée par le triangle de Pascal :
$$\left(\begin{array}{c} n \\ k \end{array} \right) = \left(\begin{array}{c} n-1 \\ k-1 \end{array} \right) + \left(\begin{array}{c} n-1 \\ k \end{array} \right)$$
3. La formule de récurrence : $\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n}{k} \left(\begin{array}{c} n-1 \\ k-1 \end{array} \right)$

Estimer la complexité pour chacune de ces méthodes. Discuter les éventuels inconvénients et avantages de chaque méthode. Illustrer votre discussion. Laquelle semble la plus efficace ?

Exercice 3

On rappelle la définition de la suite de Fibonacci $(F_n)_{n\in\mathbb{N}}$:

$$F_0 = 0, F_1 = 1, \quad \text{et } \forall n \in \mathbb{N},\ F_{n+2} = F_{n+1} + F_n.$$

On se propose d'étudier plusieurs méthodes pour calculer F_n pour de très grandes valeurs de n.

1. Quelle complexité donne la méthode naïve suggérée par la relation de récurrence ?
2. On écrit matriciellement :
$$\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix} = C \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}$$
Proposer une méthode efficace pour calculer C^n. En estimer la complexité.

3. Ecrire une fonction qui donne la valeur de F_n avec une méthode modulaire (utilisation du théorème des restes chinois). La comparer sur de grands nombres avec les précédentes.

4. Comparer les trois méthodes.

Exercice 4

Illustrer de facon expérimentale la complexité d'un algorithme que vous choisissez en utilisant une régression linéaire.

Exercice 5

Un exemple de problème NP-complet : Le voyageur de commerce. On se donne une matrice $M = (m_i, j)$ à coefficients dans $\mathbb{N}$. Etant donné un entier naturel N, peut-on trouver une permutation σ telle que $\sum_i m_{\sigma i,\sigma i+1} \leq N$.

On propose deux heuristiques qui utilise des modifications locales de la forme : On part d'une permutation σ . si on trouve i, j tels que : $m_{\sigma i,\sigma i+1} + m_{\sigma j,\sigma j+1} \geq m_{\sigma i,\sigma j+1} + m_{\sigma j,\sigma i+1}$ alors on modifie la permutation en conséquence.

Dans la première méthode, on choisit à chaque étape le couple (i, j) qui fournit la meilleure "diminution". Dans la seconde, on choisit à chaque étape le couple (i, j) au hasard parmi ceux qui conviennent.

Comparer expérimentalement les deux méthodes.