TP N°5
Algorithmes itératifs vs récursifs

Exercice1 : Suite de Fibonacci

On considère la834 suite de Fibonacci $(U_n), n \in \mathrm{N}$ définie par :

$$\begin{cases} U_0 = 0;\ U_1 = 1 \\ \forall n \geq 2: U_n = U_{n-1} + U_{n-2} \end{cases}$$

a. Ecrire une fonction *Fibo_Rec* qui calcul récursivement le n[ième] terme de la suite de Fibonacci. Donner sa complexité.
b. Ecrire une fonction *Fibo_iter* qui calcul itérativement le n[ième] terme de la suite de Fibonacci de complexité linéaire.
c. Mesurer les temps d'exécution T pour l'échantillon des nombres n ci-dessous et compléter le tableau :

n	5	10	15	20	25	35	45	60	100
T(n) de Fibo_Rec									
T(n) de Fibo_iter									

d. Comparer la complexité théorique et les mesures expérimentales des deux fonctions. Les prédictions théoriques sont-elles compatibles avec les mesures expérimentales ?
e. Représenter avec un graphe les variations du temps d'exécution T(n) de *Fibo_Rec* et *Fibo_iter*. Que constatez-vous ?
f. On peut monter que la suite (U_n), $n \in \mathrm{N}$ vérifie $U_{n+\infty} \sim \phi^n$, *où ϕ est le nombre d'or*. Ainsi $\lim_{n\to+\infty} \frac{U_n}{U_{n-1}} = \phi$. Faites une copie de la fonction Fibo_iter et modifiez-la pour tester cette propriété (utilisez le type double). A chaque itération, la fonction doit afficher U_n et le rapport $\frac{U_n}{U_{n-1}}$.

Remarque : $\phi = \frac{1+\sqrt{5}}{2} \cong 1{,}6180339887498948482045868343656$

Exercice 2 : Tours de Hanoï

"Les tours de Hanoi" est un problème classique en informatique. Il montre la puissance et la lisibilité des algorithmes définis de façon récursive. C'est un jeu de réflexion qui consiste à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire » et ceci en un minimum de coups[1], tout en respectant les 2 règles suivantes :

[1] On note qu'un jeu à 64 disques requiert un minimum de 2^{64}-1 déplacements. En admettant qu'il faille 1 seconde pour déplacer un disque, ce qui fait 86 400 déplacements par jour, la fin du jeu aurait lieu au bout d'environ 213 000 milliards de jours, ce qui

a. On ne peut déplacer plus d'un disque à la fois ;
b. On ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

1. Ecrire un algorithme récursif qui résout le problème des tours de Hanoi. On suppose qu'il y a N disques à transférer (N est un entier naturel, N>=1).
2. Calculer la complexité de cet algorithme.
3. Ecrire avec le langage C le programme correspondant.
4. Mesurer les temps d'exécution T (en secondes) pour un échantillon de données de la variable N (taille du tableau Tab) et représenter les résultats sous la forme d'un tableau :

N	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
T(s)																							

N	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	…	…	…	64
T(s)																			

5. Représenter par un graphe les variations du temps T par rapport aux valeurs de N.

g. Comparer la complexité théorique et les mesures expérimentales.
Les prédictions théoriques sont-elles compatibles avec les mesures expérimentales ?

h. Soit l'algorithme itératif du problème Tours de Hanoï. Ecrire le programme correspondant puis calculer les temps d'exécution (même tableau que la question 4. à remplir) et représenter les variations du temps par un graphe.

i. Comparer les versions récursive et itérative. Que constatez-vous ?

Equivaut à peu près à 584,5 milliards d'années, soit 43 fois l'âge estimé de l'univers (qui est 13,7 milliards d'années selon certaines sources) [source wikipédia].