Université de Sidi Bel Abbès
Département d'Informatique
Master1 2025/2026

Algorithmique et Complexité
TP1 : Algorithmes pour le Calcul des termes d'une suite

Considérons la suite d'entiers positifs définie par :

$$\begin{cases} F(0) = F(1) = 1 \\ F(n) = F(n-1) + F(n-2) \quad \text{pour } n > 1 \end{cases}$$

Ses termes sont :

1, 1, 2, 3, 5, 8, F(80)= 37889062373143906,F(100)= 1298777728820984005

Partie 1 : Implémentation avec une approche itérative

Question 1 :

Ecrire une classe java, avec **3 méthodes itératives** différentes permettant de calculer le $n^{ième}$ terme de la suite. Pour le stockage des termes :

1. La première méthode utilisera **un tableau de taille n+1.**
2. La deuxième méthode utilisera **3 variables.**
3. La troisième méthode utilisera uniquement **2 variables.**

L'utilisateur devra exécuter le programme pour différentes valeurs de n=0,.., 10, …, 80,…100,…

Question 2 :

Pour chaque méthode, afficher le temps d'exécution en millisecondes nécessaire au calcul de chaque terme. Pour cela l'instruction **System.nanoTime()** peut être utilisée. Essayer **System.currentTimeMillis().**

Partie 2 : Implémentation avec une approche récursive

Question 1 :

Compléter la classe déjà écrite dans la partie 1, avec 2 autres **méthodes récursives** et exécuter le programme pour différentes valeurs de n=0,1,.., 10, …, 80,…100,…

1. La première utilisera une approche avec **1 seul appel récursif (récursivité terminale).**
2. La deuxième utilisera une approche avec **2 appels récursifs (récursivité non terminale).**

Question 2 :

Pour chaque méthode afficher le temps d'exécution nécessaire au calcul de chaque terme.
Que peut-on remarquer sur la deuxième méthode (pour n=80, 100,….) ? Donner une explication.

Question 3 :

Parmi les 5 solutions laquelle est la plus efficace et laquelle est la moins efficace ?