TD Maple : suite de Fibonacci

► Nous nous proposons d'étudier différentes méthodes pour calculer le terme d'indice n de la suite de Fibonacci ; rappelons que cette suite est définie par $F_0 = 0$, $F_1 = 1$ et la relation de récurrence $F_n = F_{n-1} + F_{n-2}$ pour $n \geqslant 2$.

► La traduction mot à mot de cette définition donne une première version :

```
fib1 := proc(n);
 if n=0 then 0
 elif n=1 then 1
 else fib1(n-1)+fib1(n-2)
fi end;
```

Q1 Saisissez cette fonction dans votre feuille de calcul, puis affichez la séquence $(F_n)_{0\leqslant n\leqslant 10}$.

Q2 Demandez la valeur de F_{100} : que se passe-t-il ?

► Nous alons expliquer le phénomène. Notons c_n le coût du calcul de F_n, exprimé en nombre d'appels de la fonction `fib1` ; ainsi, $c_0 = c_1 = 1$ et $c_2 = 3$.

Q3 Lisez l'aide sur `trace`. Utilisez cette possibilité de Maple pour obtenir c_3 et c_4.

Q4 Montrez que la suite $(c_n)_{n\in\mathbb{N}}$ vérifie une relation de récurrence linéaire du second ordre ; donnez alors une expression *très simple* de c_n.

Q5 En vous basant sur une mesure du temps de calcul de F_{25} (par exemple), donnez une estimation du temps de calcul de F_{100}. Vous exprimerez le résultat en millions d'années.

Q6 La formulation précédente est inefficace puisque, pour calculer F_n, on calcule F_{n-2} après avoir calculé F_{n-1} : or F_{n-2} avait déjà été obtenu au cours de ce calcul ! Rectifiez ceci : rédigez une fonction `fib2` utilisant `option remember` et affichez F_{100}.

Q7 Avec cette méthode, quel est le coût du calcul de F_n si aucun terme n'a été calculé auparavant ?

Q8 Quelle critique formuleriez-vous à l'égard de cette méthode ?

► Nous allons voir qu'on peut faire bien mieux. Notons A la matrice $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$.

Q9 Saisissez cette matrice dans votre feuille de calcul et calculez A^n pour $1 \leqslant n \leqslant 5$. Quelle formule voyez-vous apparaître ?

Q10 Prouvez que cette formule est vraie pour tout n.

► Nous voyons que, pour obtenir F_n, il suffit de calculer A^{n-1}.

Q11 Montrez que ceci peut être réalisé en n'utilisant que deux matrices : l'une gardera la valeur de A, l'autre prendra pour valeurs successives les puissances de A, de l'exposant 2 à $n-1$ inclus.

Q12 Déterminez avec précision le nombre d'opérations (additions et multiplications) que requiert cette méthode.

Q13 Rédigez en Maple une fonction `fib3` mettant en œuvre cette méthode.

► Cette méthode requiert un espace mémoire constant, ou encore $\mathcal{O}(1)$, et un temps $\mathcal{O}(n)$. Nous allons voir que sur ce deuxième point, on peut encore faire mieux.

Q14 Remarquons que $A^{2n} = (A^n)^2$ et $A^{2n+1} = A^{2n} \cdot A$. Avec ces formules, combien de produits matriciels suffisent pour obtenir A^{37} ?

Q15 Plus généralement, montrez que le coût du calcul de A^n avec ces formules est égal à $\lfloor \lg(n) \rfloor + \nu(n)$ produits matriciels. Ici, lg désigne le logarithme en base 2 : $\lg(n) = \dfrac{\ln(n)}{\ln(2)}$; et $\nu(n)$ désigne le nombre de chiffres 1 dans l'écriture de n en base 2 : par exemple, $\nu(37) = 3$.

Q16 Utilisez cette idée pour rédiger une fonction `fib4`, évaluant F_n en un temps $\mathcal{O}\big((\lg(n)\big)$.

Q17 Rédigez en Maple des fonction `lg` et `nu` qui calculent $\lg(n)$ et $\nu(n)$ pour n fixé, en n'effectuant que des opérations élémentaires (additions, soustractions, calcul du quotient et/ou du reste dans la division par 2).