Université Paris Dauphine
IUP Génie Mathématique et Informatique
IUP1 – Année 2001-2002
Algorithmique et structures de données

Devoir à rendre le 25 octobre 2001

La suite de Fibonacci F_n est définie par récurrence par :

$$\begin{cases} F_0 = 1, \\ F_1 = 1, \\ F_n = F_{n-1} + F_{n-2} \quad \text{si } n > 1. \end{cases}$$

Question 1 :

Écrire une fonction récursive en langage C "`int fibo (int n)`" qui retourne le $n^{\text{ième}}$ nombre de Fibonacci F_n.

Question 2 :

Soit $T(n)$ la complexité de la fonction `fibo`. Donner l'équation de récurrence que satisfait $T(n)$ (on ne cherchera pas à résoudre cette équation).

Question 3 :

Donner une version itérative de coût linéaire $\mathcal{O}(n)$ qui retourne le $n^{\text{ième}}$ nombre de Fibonacci F_n.

Les questions 4, 5 et 6 se font sur ordinateur : vous devez taper, compiler et exécuter un programme en langage C. Pour chacune de ces questions, on imprimera le listing complet du programme demandé, ainsi que le listing du résultat du programme.

Question 4 :

Le programme suivant utilise la fonction récursive de la question 1 pour afficher les `N` premiers nombres de Fibonacci (le corps de la fonction `fibo` est à compléter par l'algorithme trouvé à la question 1) :

```
#include <stdio.h>

int fibo (int n)
{
  ...
}

void main()
{
  int i ;
  int N=40 ;
  for (i=1 ;i<=N ;i++)
    printf ("Fibo(%i) = %i \n", i, fibo(i)) ;
}
```

Donner le listing complet ainsi que le résultat de l'exécution du programme.

La valeur maximale `N` a été choisie de manière à ce que le programme ne soit pas trop lent sur mon ordinateur ; vous pouvez l'adapter à votre ordinateur.
Dites quelle valeur de `N` vous avez choisi.
Pour quelles valeurs de `n` `fibo(n)` donne-t-il une réponse instantanée sur votre ordinateur ? À partir de quelle valeur de `n` le temps de réponse dépasse-t-il quelques secondes sur votre ordinateur ?

Attention ! Sur la plupart des compilateurs C actuels, un entier de type `int` ne peut pas dépasser 2147483647 ; cela signifie qu'on ne peut calculer `fibo(n)` que jusqu'à `n`=45 ; pour `n`>45, les résultats deviennent fantaisistes. Encore une fois, ces valeurs dépendent de votre compilateur.

Question 5 :

Refaire la question 4 en remplaçant la fonction `fibo` récursive par la version itérative trouvée à la question 3. Le résultat du programme doit, bien évidement, être le même que dans la question 4.

Donner le listing complet ainsi que le résultat de l'exécution du programme.

Quelle est la vitesse de ce programme, comparée à la vitesse du programme de la question 4 ?

Question 6 :

Dans cette question, on utilise les fonctions `time` et `difftime` de la bibliothèque `<time.h>` pour mesurer le temps d'exécution de la version récursive de `fibo`.
On utilisera pour cela le programme suivant (le corps de la fonction `fibo` est à compléter par l'algorithme trouvé à question 1) :

```
#include <stdio.h>
#include <time.h>

int fibo (int n)
{
  ...
}

void main()
{
  int i ;
  time_t t0,t1 ;
  for (i=30 ;i<=45 ;i++) {
    t0 = time(NULL) ;
    fibo(i) ;
    t1 = time(NULL) ;
    printf ("i=%i fibo=%fs \n", i, difftime(t1,t0)) ;
  }
}
```

Donner le listing complet ainsi que le résultat de l'exécution du programme.