

TD de Algorithmique n° 6**10 Novembre 2011****(Correction)****I) Suite de Fibonacci**

On considère la suite

$$u_{n+2} = u_{n+1} + u_n$$

et nous allons voir que comment la calculer importe beaucoup...

Exercice 1 :

Comment l'écriveriez vous simplement ? Dessinez l'arbre des appels récursifs pour `fibonacci`
5. Quelle est la complexité ?

Exercice 2 :

Si maintenant nous stockons les 2 dernières valeurs, quel est l'algorithme ? Quel est son coût ?

Exercice 3 :

On peut remarquer que

$$\begin{pmatrix} u_{n+1} \\ u_{n+2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix}$$

comment en tirer profits pour calculer plus efficacement ?

II) Justesse et terminaison**Exercice 4 :**

Quel est le problème avec le code qui se trouve dans le balladeur MP3 d'un vendeur de fenêtre :

```
1 year = ORIGINYEAR; /* = 1980 */
2 while (days > 365)
3     do if (IsLeapYear(year))
4         do if (days > 366)
5             do
6                 days -= 366;
7                 year += 1;
8     else
9         do
10            days -= 365;
11            year += 1;
```

Que doit il faire ? Est il correct ?

Exercice 5 :

Montrer formellement que la recherche du d'un tableau est correct et terminant.

Exercice 6 :

Prouvons la correction et la terminaison du tri rapide

PARTITION(A, g, d)

```
1  pivot ←  $A[g]$ 
2   $i$  ←  $g - 1$ 
3   $j$  ←  $d + 1$ 
4  while true
5      do repeat  $j$  ←  $j - 1$ 
6          until ( $A[j] \leq pivot$ )
7      repeat  $i$  ←  $i + 1$ 
8          until ( $A[i] \geq pivot$ )
9      if ( $i < j$ )
10         then echange  $A[i] \leftrightarrow A[j]$ 
11         else return  $j$ 
```

TRI-RAPIDE-R(A, g, d)

```
1  if ( $g < d$ )
2      then  $p$  ← PARTITION( $A, g, d$ )
3          TRI-RAPIDE-R( $A, g, p - 1$ )
4          TRI-RAPIDE-R( $A, p + 1, d$ )
```

TRI-RAPIDE(A)

```
1  TRI-RAPIDE( $A, 1, \text{LENGTH}(A)$ )
```