

Algorithmique TD4 Correction: Récursivité

4 novembre 2008

1. PGCD

- (a) Ecrire une fonction récursive permettant de calculer le PGCD

```
fonction pgcd(a: entier , b: entier) : entier
  si a < b alors
    retourner pgcd(a, b-a)
  sinon si a = b alors
    retourner a
  sinon
    retourner pgcd(a-b, b)
  fsi
finfonction
```

2. Affichage d'un nombre en binaire

- (a) Ecrire une procédure récursive affichant un nombre en binaire à partir de sa valeur décimale

```
procedure affichebinaire(n: entier) : entier
  si (n mod 2) ≠ 0 alors
    affichebinaire(n div 2)
  fsi
  ecrire(n mod 2)
finprocedure
```

- (b) Donner la trace de l'appel suivant : `affichebinaire(12)`
(c) Comment écrire un algorithme itératif produisant exactement le même résultat
Une solution est de stocker les valeurs dans un tableau puis de le parcourir dans le bon ordre pour l'afficher.

3. Suite de Fibonacci

- (a) Ecrire une version récursive de Fibonacci de complexité linéaire

```
procedure fibrec(E n: entier , S a: entier , S b: entier)
variables
  x: entier
```

```

y: entier
si n=0 alors
  a ← 1
  b ← 0
sinon
  fibrec(n-1,x,y)
  a ← x+y
  b ← x
fsi
finprocedure
fonction fib(E n: entier): entier
variables
  a: entier
  b: entier
  fibrec(n,a,b)
  retourner a
finfonction

```

4. Tours de Hanoï

Soit le programme suivant :

```

procedure deplacer(n: entier , de: entier , a: entier , par: entier )
  si n > 0 alors
    deplacer(n-1,de,par,a)
    bouger(de,a)
    deplacer(n-1,par,a,de)
  fsi
finprocedure

```

avec la procédure **bouger(de,a)** qui déplace un disque de la colonne **de** vers la colonne **a**

- (a) Considérons que l'on a donc trois colonnes disponibles (**de,a,par**) dont la première contient n disques superposés par diamètres décroissant et les deux autres ne contiennent aucun disque. Dessiner le résultat de l'appel suivant de la procédure déplacer : **deplacer(4,1,2,3)**
- (b) Evaluer la complexité de cette procédure
 $C(n) = 2 * C(n - 1) + cst$ donc complexité exponentielle ($O(2^n)$)
- (c) Peut-on trouver facilement une version itérative de cette procédure
 Non car récursivité non terminale

5. Rendu de monnaie

Considérons que l'on veuille rendre la monnaie sur une somme $N \leq 100$ pour un achat de valeur V ($V \leq N$). Ecrire un algorithme qui détermine les différentes possibilités pour rendre la monnaie sachant que l'on dispose des valeurs de pièces et billets suivantes : 1,2,5,10,50.