

## J1MI2013: Algorithmes et Programmes: feuille 9

### Réversivité

#### Travaux pratiques

Récupérer depuis le site de l'UE l'archive `tp09.tar.gz` et désarchivez-le. Placez vous ensuite dans le répertoire `tp09`.

**Exercice 1.** Dans le fichier `testFonctionsRec.c` écrire et appeler les fonctions de l'exercice 3 de la feuille de TD 9.

**Exercice 2.** Les fichiers `testFiboRec.c` et `testFiboIter.c` contiennent respectivement les versions récursive et itérative de la fonction qui calcule le  $n$ -ième terme ( $n \geq 0$ ), de la suite de Fibonacci<sup>1</sup>. Sous Unix il est possible d'afficher le temps de calcul nécessaire pour exécuter une commande en utilisant la commande `time`. Ainsi, la commande :

```
time ./testFiboRec 10
```

affiche le terme de rang 10 de la suite de Fibonacci et trois valeurs de temps en secondes :

```
fib(10)= 89
```

```
real 0m0.003s
```

```
user 0m0.004s
```

```
sys 0m0.000s
```

La valeur à retenir est la deuxième, elle représente le temps de calcul nécessaire.

Comparer les temps affichés pour chacune des deux versions pour les valeurs suivantes :

$n = 10$ ,  $n = 30$ ,  $n = 40$ ,  $n = 45$ .

Attention : au delà de  $n=45$  on n'obtient pas un résultat correct car le nombre de Fibonacci que l'on veut calculer dépasse la capacité de stockage pour une variable de type `int` qui ne peut représenter que les entiers entre  $-2^{31}$  et  $2^{31} - 1$ .

**Exercice 3.** Dans le fichier `testCroissRec.c`, écrire une fonction récursive

`bool estStrictCroissantRec(int t[], int n)` qui renvoie `true` si le tableau `t` de `n` entiers est strictement croissant, et `false` sinon.

Par convention, un tableau contenant 0 ou 1 élément est considéré comme strictement croissant.

**Exercice 4.** Dans le fichier `testPalinSectionRec.c`, écrire une fonction récursive

`bool estPalindromeSectionRec(int t[], int i, int j)` qui renvoie `true` si la section du tableau `t` entre les indices `i` et `j` inclus, est un palindrome et renvoie `false` sinon. Par convention, si  $i > j$  la fonction renvoie `true`.

---

1. pour la définition voir exercice 7 de la feuille de TD 2

Exemple : Si  $t=[3,7,2,1,1,2,5]$  l'appel `estPalindromeSectionRec(t, 2, 5)` doit renvoyer `true`, alors que l'appel `estPalindromeSectionRec(t, 0, 6)` doit renvoyer `false`.

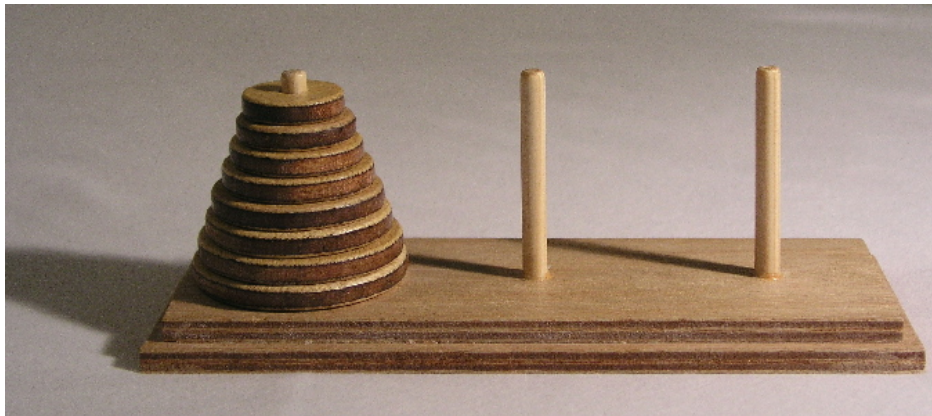
**Exercice 5.** La fonction itérative `int SyracTemps(int n)` (demandée à l'exercice 2 de la feuille 3 de TD machine) renvoie le *temps de vol* de la suite de Syracuse de  $n$ . Écrire dans le fichier `testSyracuseRec.c` une version récursive de cette fonction.

Exemple : le temps de vol pour  $n = 10$  est 6 car 10 est le terme de rang 0 de la suite de Syracuse : 10, 5, 16, 8, 4, 2, 1 et 1 est donc le terme de rang 6.

**Exercice complémentaire 1.** Dans le fichier `testMultAlexRec.c`, écrire une fonction récursive pour implémenter l'algorithme de la *multiplication alexandrine* définie dans l'exercice complémentaire de la feuille 3 de TD.

**Exercice complémentaire 2.** Le jeu des tours de Hanoi se joue avec 3 piquets, appelés *d* (comme départ), *a* (comme arrivée), et *i* (comme intermédiaire), et  $n$  disques de tailles différentes que l'on met sur ces piquets.

On part de la configuration où les  $n$  disques sont disposés sur le piquet *d* de départ, comme sur la photo suivante (avec  $n = 8$ ).



L'objectif est de déplacer les disques du piquet *d* vers le piquet *a*, en déplaçant un seul disque à chaque étape. La contrainte à respecter est qu'on n'a pas le droit de placer un disque sur un disque plus petit que lui.

1. En supposant que vous savez déplacer les  $(n - 1)$  disques les plus petits d'un piquet vers un autre en utilisant une suite de mouvements, comment faire pour déplacer les  $n$  disques du piquet *d* vers le piquet *a* ?
2. Décrire un algorithme récursif permettant de résoudre le jeu, en fournissant la suite des déplacements.
3. Combien cet algorithme fait-il de déplacements de disques ?
4. Dans le fichier `testHanoi.c`, écrire une fonction qui implémente cet algorithme. La fonction affichera (par l'instruction `printf`) la suite des déplacements.