UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

# FLMA607 – Calcul formel – 2010-2011

## TP 1 : Premiers algorithmes (deux séances)

### Exercice 1 (Multiplication de polynômes)

1) Implémenter une fonction réalisant la multiplication de deux polynômes par l'algorithme naïf. On représentera les polynômes sous forme de vecteurs (`vector`, package `linalg`). La longueur d'un vecteur est donnée par `vectdim`.
2) Écrire une procédure récursive qui réalise la multiplication de deux polynômes par l'algorithme de Karatsuba (mêmes consignes).
3) Comparer les temps de calcul pour quelques polynômes pris au hasard (on pourra utiliser les commandes `time();` et `randvector`) de degré inférieur à 1000. Quel algorithme est le plus rapide ? Commentez.
4) Vérifier expérimentalement que le temps du calcul du produit de deux polynômes de degré $n$ est en $O(n^2)$ par le premier algorithme et en $O(n^{1,59})$ par le second. On commencera par remplir un tableau contenant, pour suffisament de valeurs de $n$, les temps de calcul des deux algorithmes. On pourra utiliser la fonction `pointplot` pour visualiser les choses. On pourra utiliser le logarithme pour obtenir une droite.

### Exercice 2 (Test de Fermat)

1) Soit $p = 2456734573$. Vérifier que $p$ est premier à l'aide de la commande `isprime` de Maple. On souhaite calculer $2^{p-1} \mod p$. Quel résultat attend-on ? Dans Maple, exécuter les deux lignes de commandes suivantes et expliquer le calcul effectué.

```
> 2^(p-1) mod p ;
> 2&^(p-1) mod p ;
```

2) Écrire un programme `TestFermat` qui prend en entrée deux entiers $n$ et $a$ et qui rend en sortie `true` si $n$ passe le test de Fermat en base $a$ (c'est-à-dire vérifie $a^{n-1} \equiv 1 \mod n$) et `false` sinon.
3) Trouver le plus petit entier $N$ impair composé qui passe le test de Fermat en base 2.
4) Écrire un programme `TestFermatkfois` qui prend en entrée deux entiers $n$ et $k$ et qui rend `true` si $n$ passe le test de Fermat en base $a$ pour les $k$ premiers entiers $a$ qui sont premiers à $n$, `false` sinon.
5) Écrire un programme `TestFermatjusquek` qui prend en entrée deux entiers $n$ et $k$ et qui rend `true` si $n$ passe le test de Fermat en base $a$ pour tout entier $a$ inférieur à $k$ et premier à $n$, `false` sinon.
6) Trouver le plus petit entier $n$ composé impair qui passe le test de Fermat en base $a$ pour tout entier $a$ inférieur à $n$ et premier à $n$. L'entier $n$ est le plus petit nombre de Carmichael. Trouver le deuxième nombre de Carmichael.
7) Calculer `TestFermatkfois`$(n, k)$ où $k = 10$ et $n = 727993807201$, $n = 25342565819$. Comparer avec la procédure `isprime`. Conclusion ?

### Exercice 3 (Les « mauvaises bases » du test de Fermat)

Soit $n$ un entier composé. On appelle indicateur d'Euler de $n$ et on note $\varphi(n)$ le nombre d'entiers non nuls inférieurs à $n$ et premiers à $n$.
1) Écrire une procédure `IndicEuler` qui prend en entrée un entier $n$ et calcule le nombre d'entiers entre 2 et $n-1$ premiers avec $n$. Comparer avec la commande `numtheory[phi](n)` de Maple.
2) Écrire une procédure maple, `MauvaisA` qui prend en entrée un entier $n$ et rend en sortie le nombre d'entiers $a$ compris entre 2 et $n-1$ et premiers à $n$ pour lesquels $n$ passe le test de Fermat en base $a$.
3) À l'aide de Maple, construire la liste des couples $\left[n, \frac{\texttt{MauvaisA}(n)}{\texttt{IndicEuler}(n)}\right]$ pour $n$ entier composé inférieur ou égal à 1000 (ne pas afficher la liste) puis représenter les points obtenus sur un graphe. Commentez.