

# Preuve et Analyse des Algorithmes

## Examen 2h

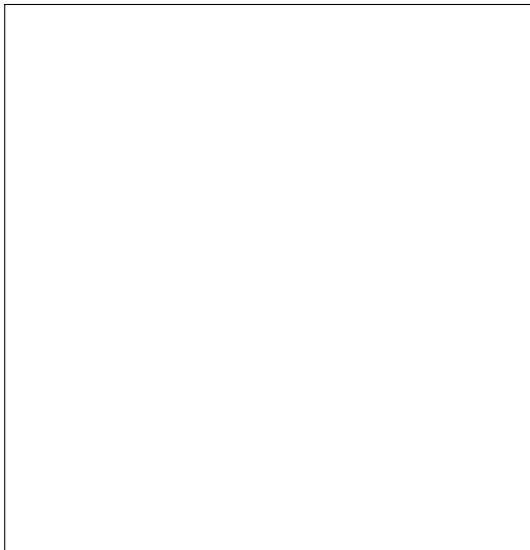
8 janvier 2010

### 1 Suite récurrente

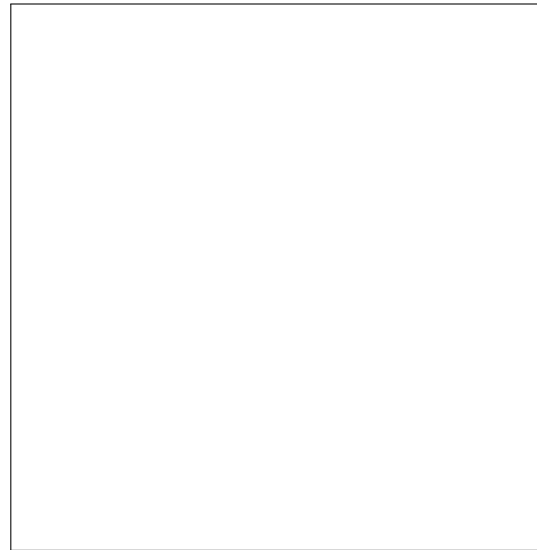
On considère la suite récurrente  $R_n$  définie par les relations :  $R_0 = 0$ ,  $R_1 = 1$ ,  $R_2 = 2$  et

$$\forall n \geq 3, \quad R_n = R_{n-1} - R_{n-2} + R_{n-3}$$

- (a) Ecrire un algorithme itératif pour calculer  $R_n$ .



- (c) Donner une implantation en langage C de cet algorithme itératif.



### 2 Analyse

- (b) Quel est la forme du temps de calcul de cet algorithme ?

- (a) Ecrire un algorithme récursif pour calculer  $R_n$ .

On note  $T(n)$  le temps de calcul de cet algorithme récursif.

(b) Montrer que pour  $n \geq 3$ , on a :

$$T(n) \geq 3T(n-3).$$

(c) En déduire

$$T(n) = \Omega(3^{n/3}).$$

(d) Comparer l'efficacité de la version itérative à celle de la version récursive.

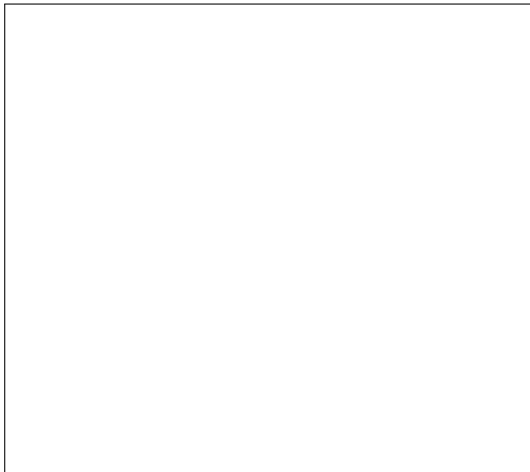
### 3 Euclide à la main

Résoudre le problème de Bâchet-Bézout pour les entiers  $a = 120$ , et  $b = 23$ .

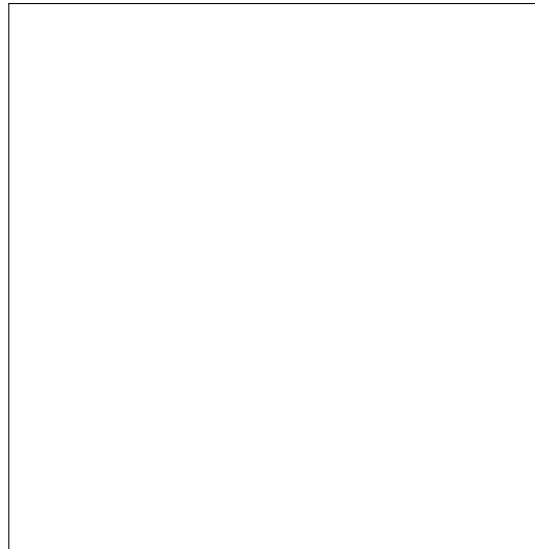
Quel est l'inverse de 23 modulo 120 ?

### 4 Nombre

On représente les nombres par des tableaux de  $N$  chiffres en base  $B$ . Ecrire un algorithme pour décrémenter un nombre.



(b) Ecrire un algorithme naïf qui prend en paramètre un tableau de dimension  $n$  et renvoie vrai si  $T$  possède un élément majoritaire, et faux sinon.



## 5 Invention

Soit  $T$  un tableau de  $n$  objets. Un objet  $x$  est dit majoritaire dans  $T$  si le nombre d'occurrences de  $x$  dans  $T$  est strictement supérieur à  $n/2$ .

(a) Un tableau possède au plus un objet majoritaire. Pourquoi ?



(c) Préciser le temps de calcul en fonction de  $n$ . Dans le meilleur des cas, dans le pire des cas.



(c) Comment améliorer cet algorithme ?