MATH EN INFO

FICHE EXPLICATION Nº 10

Modulo

1 Descriptif

L'objectif de ce défi est d'implémenter une méthode renvoyant le reste obtenu dans la division euclidienne du nombre binaire courant par le nombre binaire donné en paramètre. Pour cela, il nous faut implémenter une division euclidienne en binaire. Pour cela, nous allons simplement poser la division comme vous avez appris à le faire en primaire.

Supposons que l'on cherche à calculer a=110001 modulo b=101. On initialise deux variables $r = a$ et $q = 0$ qui vont représenter le reste et le quotient de notre division.

1. Soit n le décalage nécessaire pour que r et b est la même taille. Ici $n = 3$.
2. On calcule b', b décalé de n bits. Ici $b' = 101000$.
3. Si $b' > r$ alors on remplace b' par b décalé de $n-1$ bits puis n par $n-1$. (Pas besoin dans notre cas).
4. On remplace r par $r - b'$. Dans notre cas r devient 1001.
5. On ajoute 2^n à q. Dans notre cas q devient 1000.
6. On recommence à l'étape 1 jusqu'à ce que $r < b$.

Si on effectue un tour de plus dans notre cas, on obtient :

1. $r = 1001$, $b = 101$, donc $n = 1$.
2. $b' = 1010$.
3. $b' > r$ donc $b' = 101$ et $n = 0$.
4. $r = r - b' = 100$.
5. On ajoute 2^0 à $q = 1000$. Donc q devient 1001.
6. $r < b$ donc on s'arrête.

$r = 100$ est le modulo de a par b et $q = 1001$ est le quotient de a par b.

2 Protocole

1. Une fois la connexion établie, le serveur commence par envoyer un premier message annonçant le début du défi :

   -- Début du défi : Modulo --

   Ce message n'attend pas de réponse.
2. Le serveur envoie ensuite une série de nombres binaires (de taille aléatoire) deux par deux.
3. Pour chaque paire de nombres binaires, le serveur doit recevoir en retour un nombre binaire (sous forme binaire) égal au modulo du premier nombre par le deuxième.
4. Après chaque réponse, le serveur enverra un message commençant par "OK" ou "NOK" suivant si la réponse est correcte ou non.

5. A la fin du défi, le serveur enverra un message indiquant "Défi validé" ou "Défi échoué !". Aucune réponse n'est attendue.
6. Le serveur terminera la communication par le message "FIN", votre client devra alors fermer la socket. Aucune réponse n'est attendue.

3 Exemple de communication

Voici un exemple (incomplet) d'une communication pour ce défi. Dans cet exemple les "<" et ">" indiquent le sens de transfert de chaque message et ne doivent pas être présents dans la communication.

< -- Début du défi : Modulo --
< 1101101001011001001
< 1100001010101101010110101
> 1101101001011001001
< OK
< 10001111000100001000
< 110101110
> 101001100
< OK
< 110111110
< 100100110110110000011100
> 110111110
< OK