Mémo n. 7

Mais à quoi ça sert ?

I32 Preuves et Analyses d'algorithmes

P. Véron

Ça y est, le calvaire se termine, vous avez réussi à supporter l'ensemble des cours, td et tp de ce module et inlassablement la même phrase retentit dans votre esprit : "Mais à quoi ça sert toutes ces formules inutiles, c'est pas de l'informatique. Moi quand j'écris un programme, je l'exécute et je vois bien s'il marche ou pas!!!" Cette dernière remarque renvoie les pauvres enseignants du I32 dans le fin fond de leur désert où désespérément ils tentent de guider quelques brebis sur les voies au combien périlleuses de l'algorithmique tandis que les autres sombrent dans les méandres du bidouillage ..... Afin de pouvoir ramener quelques brebis sur le droit chemin, j'ai décidé de rédiger ce dernier petit mémo autour d'un exemple bien concret.

Le problème du calcul de x^n

Dans certaines applications cryptographiques (applications qui consistent à brouiller un message lors de sa transmission pour assurer sa confidentialité), il est parfois nécessaire de calculer x^c où c est un grand nombre (généralement de l'ordre de 1024 bits). Quel est le plus grand entier codable sur 1024 bits ? C'est $2^{1024-1} = 17976931348623159077293051907890243361797697894230657273430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137215$.
Suite à la demande du ministère de la défense, le célèbre M. Tee, cryptographe de renommée internationale, propose l'algorithme suivant pour réaliser un tel calcul :

```
1  Algorithme 1
2  données y, x, c : entiers
3    début
4      lire(x)
5      lire(c)
6      y ← 1
7      tant que c ≠ 0 faire
8                  y ← y * x
9                  c ← c − 1
10     fintq
11     ecrire(y)
12   fin
```

Etant donné l'extrême simplicité de cet algorithme, les experts du ministère sont persuadés que ce dernier réalise bien le calcul de x^c, inutile d'en donner une preuve. Ils se soucient maintenant de déterminer sur quelle machine cet algorithme doit s'exécuter pour que le calcul de x^c se réalise en au plus une seconde. Pour simplifier les choses on admettra qu'une machine dont le processeur fonctionne à N Ghz ($= N \times 10^9$ hz) est capable d'exécuter $N \times 10^9$ opérations par seconde (c'est à dire N milliards d'opérations par seconde). En réalité elle exécutera beaucoup moins d'opération car la fréquence des processeurs indique le nombre d'instructions élémentaires exécutées par seconde et une simple multiplication est composée de plusieurs instructions élémentaires.

Dans le pire des cas $c = 2^{1024} - 1$, on aura donc $2^{1025} - 2$ opérations à exécuter (qui correspondent à $p \leftarrow p * x$ et $c \leftarrow c - 1$ exécutées lors des c passages dans la boucle). Si la machine est capable d'exécuter $N \times 10^9$ opérations par seconde, on cherche donc N tel

que

$$\frac{2^{1025}-2}{N\times 10^9}\leqslant 1$$

ce qui donne

$$N\geqslant \frac{2^{1025}-2}{10^9}\simeq 10^{300}$$

La machine à utiliser doit donc fonctionner à 10^{300} Ghz !!!! Actuellement les machines les plus performantes fonctionnent à 2 Ghz ! Pourtant la cryptographie est utilisée de partout : cartes à puces, courrier électronique, commerce électronique, transactions bancaires, ... Comment cela est-il possible ?

Parallèlement à la proposition de M. Tee, un ex-étudiant de la très célèbre université de Toulon avait proposé au ministère un autre algorithme :

```
1  Algorithme 2
2  données x, y, c : entiers
3    début
4      lire(c)
5      lire(x)
6      y ← 1
7      tant que c ≠ 0 faire
8              si c est impair alors
9                                  y ← x * y
10             finsi
11             x ← x * x
12             c ← c div 2
13     fintq
14     ecrire(y)
15   fin
```

Face à ce dernier les experts du ministère sont restés très sceptiques quant au fait qu'il calculait bien x^c. Il leur fallait une preuve. Tout d'abord l'étudiant démontra que son algorithme fonctionnait en effectuant diverses exécutions pour des valeurs différentes. Ceci ne suffit pas à convaincre les experts qui voulaient être certains que l'algorithme fonctionne pour n'importe quelles valeurs de x et de c. L'enjeu était d'importance, hors de question qu'un message ultra confidentiel soit chiffré de façon erroné si cet algorithme avait des défaillances pour des valeurs particulières.
Fouillant dans les profondeurs de sa mémoire, l'étudiant se rappela de son cours sur les "Preuves de programmes" (le polycopié est parfait, coincé sous le pied du bureau, il est à la hauteur idéale pour maintenir ce dernier stable). Il démontra donc aux experts de façon formelle que l'algorithme calcule bien x^c pour toutes valeurs de x et de c en leur prouvant que

$$\{X^C = yx^c\}$$

est un invariant de boucle, C et X étant les valeurs initiales données aux variables c et x par l'utilisateur (faites le à titre d'exercice). Une fois la boucle terminée (et là aussi l'étudiant en donna une preuve) on a $c = 0$ et donc $y = X^C$!
Parfaitement convaincus les experts du ministère se posèrent alors la même question que pour le premier algorithme. Quelle devra être la puissance de la machine pour pouvoir effectuer les calculs en moins d'une seconde ? Chaque passage dans la boucle fait intervenir au plus 2 multiplications (on néglige ici la division par 2, car au niveau de la machine il s'agit d'un simple décalage de la représentation binaire de c). Le nombre de passages dans la boucle est $\lfloor \log_2 c \rfloor + 1$ (cf. mémo 0) ainsi le nombre maximum d'opérations effectuées est $2\lfloor \log_2 c \rfloor + 2$. Lorsque $c = 2^{1024} - 1$, ce nombre est donc majoré par 2050. On cherche donc N tel que :

$$\frac{2050}{N \times 10^9} \leqslant 1$$

ce qui donne

$$N \geqslant 0.205 \times 10^{-5} = 2.05 \text{ hz } !!$$

Pour $N = 2.05$ hz on obtient exactement une seconde de calcul. L'algorithme est utilisable sur des machines qui existaient il y a plus de vingt ans ! A titre de comparaison, actuellement la fréquence du processeur de certaines calculatrices scientifiques est de l'ordre de 10 Mhz (10000 hz).

Conclusion : M. Tee devrait retourner faire un tour sur les bancs de l'université !