Pgcd, classique ou tropical (Denise Vella-Chemla, 29.4.2016)

On peut programmer la fonction qui renvoie le pgcd de deux nombres selon l'algorithme d'Euclide par :

```
#include <cmath>

int pgcd(int m, int n) {
  while (m != 0) { int r ;
    r = n % m ;
    n = m ;
    m = r ;
  }
  return(n) ;
}
```

Mais on peut également programmer cette fonction ainsi, en remplaçant l'addition habituelle par le max et la multiplication habituelle par la soustraction, comme cela se fait en algèbre tropicale. On remplace n par $n \oplus (n \oslash m)$ et m par $n \oslash (m \oplus (n \oslash m))$

```
#include <cmath>

int pgcd(int m, int n)
{
  if (m >= n) {int r ; r = m ; m = n ; n = r ;}

  while (m != 0) { int r ;
    r = n-max(m, n-m) ;
    n = max(m, n-m) ;
    m = r ;
  }
  return(n) ;
}
```

Cependant le premier petit échange de variables au début de la fonction laisse à désirer.

Cette idée permet-elle de préciser un peu la connaissance qu'on a de l'ensemble des nombres premiers ? On imagine l'ensemble $\mathbb{N}^*$. On applique la fonction *pgcd* à toute paire d'entiers, cette fonction munit l'ensemble des entiers d'une structure d'ordre partiel (1 est plus petit que tous les autres, un diviseur d est plus petit que tous ses multiples kd, k non nul, mais deux nombres premiers entre eux ne sont pas comparables). Les nombres à distance 1 de 1 (dans le graphe de divisibilité) sont les nombres premiers, les nombres composés ont au moins un premier entre eux-mêmes et 1. L'algèbre tropicale apporte peut-être quelque chose en terme de nombre de solutions de certaines équations polynomiales tropicales.