Algorithme d'Euclide étendu

hbouia - 18 novembre 2017

Introduction :

Tout d'abord MERCI à qui a rédigé la page web :
`http://fr.wikipedia.org/wiki/Algorithme_d%27Euclide_%C3%A9tendu`
à propos de ce sujet dont je reprends le texte du début :
L'algorithme d'Euclide étendu est une variante de l'algorithme d'Euclide qui permet, à partir de deux entiers a et b, de calculer non seulement leur plus grand commun diviseur (PGCD), mais aussi un de leurs couples de coefficients de Bézout (deux entiers u et v tels que au + bv = PGCD(a, b)). Quand a et b sont premiers entre eux, u est alors l'inverse pour la multiplication de a modulo b (et v est de la même façon l'inverse modulaire de b, modulo a), ce qui est un cas particulièrement utile. L'algorithme d'Euclide étendu fournit également une méthode efficace non seulement pour déterminer quand une équation diophantienne ax+by= c possède une solution, ce que permet déjà l'algorithme d'Euclide simple, mais également pour en calculer dans ce cas une solution particulière, dont on déduit facilement la solution générale.

Code Python :

```python
# -*- coding: utf-8 -*-
"""
@author: hbouia (Created on Sun Nov 19 11:05:51 2017)
Algoritme d'Euclide étendu
Sitographie : https://fr.wikipedia.org/wiki/Algorithme_d%27Euclide_%C3%A9tendu
"""
def igcd(a,b):
    # Initialisation
    d,u,v,d1,u1,v1=a,1,0,b,0,1
    # Calcul
    while d1!=0:
        q=d//d1
        d,u,v,d1,u1,v1=d1,u1,v1,d-q*d1,u-q*u1,v-q*v1
    return (d,u,v)

a,b=488456,18546
a,b=4445847,64545454
d,u,v=igcd(a,b)
print 'pgcd(%d,%d) = %d' % (a,b,d)
print '(%d)*%d + (%d)*%d = %d' %(u,a,v,b,d)

# Exemple :
# pgcd(488456,18546) = 2
# (-317)*488456 + (8349)*18546 = 2
```

FIGURE 1 – Code Python avec exemple