Bezout's Theorem - Extended Euclidean Algorithm iterative

Arabic Animated intuition

Extended Euclidean algorithm iterative

ax+by=gcd(a,b) 1

ax_prev+by_prev= r_prev 2

ax+by= r_cur 3

gcd(r_prev,r_cur) r_prev from the first gcd call

a=r_prev, b=r_cur

r_new =r_prev-Q*r_cur 4

r_new=(ax_prev+by_prev)-q(ax+by)

a (x_prev-qx) +b (y_prev-qy)=r_new 5

from the Euclidean algorithm we know that

gcd(a,b)= gcd(b,a%b)

gcd(a,b)=gcd(b,r)

r is the remainder R

r=a-b*Q

so we are continually trading in
the gcd of a pair for
the gcd of a smaller pair.
At the last step, we have gcd

gcd(a,b) =gcd(b,a%b)

gcd(r_prev,r_cur)=gcd(r_cur,0)=r_cur

Extended Euclidean algorithm iterative

$ax+by=gcd(a,b)$ 1

ax_prev+by_prev= r_prev 2

ax+by= r_cur 3

r=a-q*b

a=r_prev, b=r_cur

r_new =r_prev-Q*r_cur 4

r_new=(ax_prev+by_prev)-q(ax+ay)

a(x_prev-qx) +b(y_prev-qy)=r_new 5

from the Euclidean algorithm we know that

gcd(a,b)= gcd(b,a%b)

gcd(r0,r1)=gcd(r1,r0%r1)

gcd(r_last,0) =r_last

gcd(r_prev,r_cur)=gcd(r_cur,0)=r_cur

a(1)+b(0)=a

a(0)+b(1)=b

x_new=x_prev-qx

y_new=y_prev-qy

gcd(a,b)=gcd(b,a%b)