GCD

2420010, RHIE, Suyeong

1. gcd (24, 36),

→ gcd (36, (24 rem 36)) ... by (gcd 2)

→ gcd (36, 24) ... by (remainder)

→ gcd (24, (36 rem 24)) ... by (gcd 2)

→ gcd (24, 12) ... by (rem)

→ gcd (12, (24 rem 12)) ... by (gcd 2)

→ gcd (12, 0) ... by (rem)

→ 12 ... by (gcd 1)

eq gcd (X, 0) = X, ... gcd 1

eq gcd (X, N₂Y) = gcd (N₂Y, X rem N₂Y), ... gcd 2

*) m rem NN = (rem m NN) = Remainder

FACT

1. fact (5),

→ 5 · fact (p 5) ... by (f2)

→ 5 · fact (4) ... by (p)

→ 5 · (4 · fact (p 4)) ... by (f2)

→ 5 · (4 · fact (3)) ... by (p)

→ 5 · (4 · (3 · fact (p 3))) ... by (f2)

→ 5 · (4 · (3 · fact (2))) ... by (p)

→ 5 · (4 · (3 · (2 · fact (p 2)))) ... by (f2)

→ 5 · (4 · (3 · (2 · fact (1)))) ... by (p)

→ 5 · (4 · (3 · (2 · (1 · fact (p 1))))) ... by (f2)

→ 5 · (4 · (3 · (2 · (1 · fact (0))))) ... by (p)

→ 5 · (4 · (3 · (2 · (1 · 1)))) ... by (f1)

→ 5 · (4 · (3 · (2 · 1))) → 120

→ 5 · (4 · (3 · 2)) = 120

→ 5 · (4 · 6)

→ 5 · 24

all by (x)

eq fact (0) = 1, ... f1

eq fact (N₂X) = N₂X · fact (p N₂X), ... f2

*) p NN = (- nn 1) = Previous

OEDC-FACT

1. oedc-fact(5),
→ g(5, 1) ... by(of 2)
→ cond(5>1, (g(5, 2·1)·g(S(5-1), 2·1)), 5)
... by(g)
→ cond(true, (g(5, 2·1)·g(S(5-1), 2·1)), 5)
... by(>)
→ (g(5, 2)·g(S(5-1), 2·1)) ... by(cond1)
→ cond(5>2, (g(5, 2·2)·g(S(5-1)), 2·2), 5)
... by(g)
→ cond(true, (g(5, 2·2)·g(S(5-2)), 2·2), 5) ... by(>)
→ (g(5, 4)·g(S(5-2), 2·2)) ... by(cond1)
→ cond(5>4, (g(5, 2·4)·g(S(5-4)), 2·4), 5) ... by(g)
→ cond(true, (g(5, 2·4)·g(S(5-4), 2·4)), 5) ... by(>)
→ (g(5, 8)·g(S(5-4), 2·4)) ... by(cond1)
→ cond(5>8, (g(5, 2·8)·g(S(5-8), 2·8)), 5) ... by(g)
→ cond(false, (g(5, 2·8)·g(S(5-8), 2·8)), 5) ... by(>)
→ (5·g(S(5-4), 2·4)) ... by(cond2)
→ g(1, 8) ... by(S(-) + (x)op)
→ cond(1>8, (g(1, 2·8)·g(S(1-8), 2·8)), 1) ... by(g)
→ cond(false, (g(1, 2·8)·g(S(1-8), 2·8)), 1) ... by(>)
→ (5·1) ... by(cond2)
→ (5·g(S(5-2), 2·2)) ... by((x)op)
→ g(3, 4) ... by(S(-) + (x)op)
→ cond(3>4, (g(3, 2·4)·g(S(3-4), 2·4))3) ... by(g)
→ cond(false, (g(3, 2·4)·g(S(3-4), 2·4))3) ... by(>)

eq oedc-fact(0) = 1. ... of 1
eq oedc-fact(N2x)
= g(N2x, 1). ... of 2
eq g(X, Y) = cond(X > Y,
g(x, 2·Y)·g(S(X-Y), 2·Y),
X), ... g
eq cond(true, X, Y) = X ... cond1
eq cond(false, X, Y) = Y ... cond2

→ (5 · 3) ··· by (cond 2)

→ (15 · g(sd(5, 1), 2 · 1)) ··· by (x) op)

→ g(4, 2) ··· by (sd + (x) op)

→ Cond(4 > 2, (g(4, 2 · 2) · g(sd(4, 2), 2 · 2)), 4) ··· by (g)

→ Cond (true, (g(4, 2 · 2) · g(sd(4, 2), 2 · 2)), 4) ··· by (>)

→ (g(4, 4) · g(sd(4, 2), 2 · 2)) ··· by (cond 1) + (x) op)

→ Cond(4 > 4, (g(4, 2 · 4) · g(sd(4, 4)), 2 · 4), 4) ··· by (g)

→ Cond (false, (g(4, 2 · 4) · g(sd(4, 4)), 2 · 4), 4) ··· by (>)

→ (4 · g(sd(4, 2), 2 · 2)) ··· by (cond 2)

→ g(2, 4) ··· by (sd + (x) op)

→ Cond(2 > 4, (g(2, 2 · 4) · g(sd(2, 4), 2 · 4)), 2) ··· by (g)

→ Cond (false, (g(2, 2 · 4) · g(sd(2, 4)), 2 · 4), 2) ··· by (>)

→ (4 · 2) ··· by (cond 2)

→ (15 · 8) ··· by (x) op)

→ 120