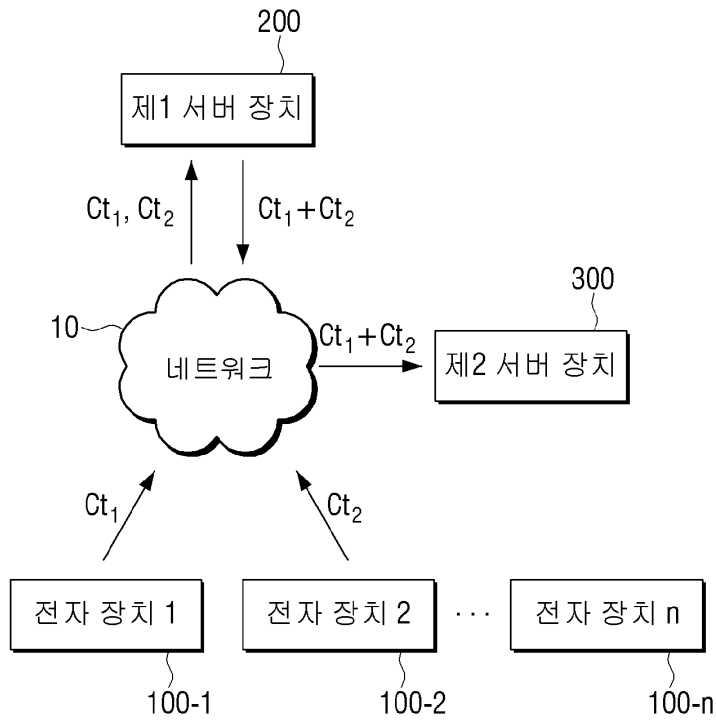
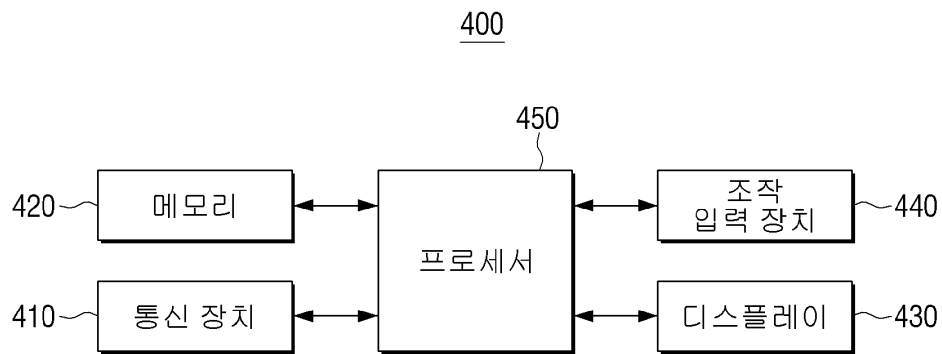


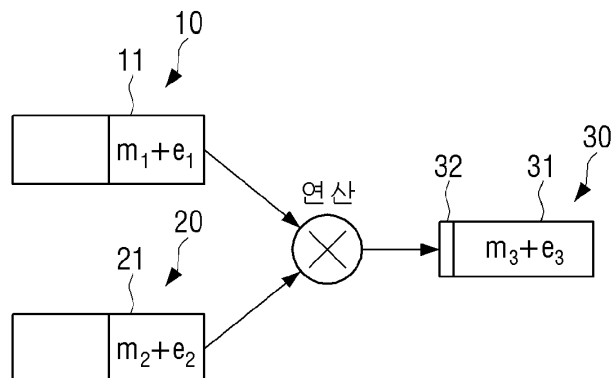
[도1]



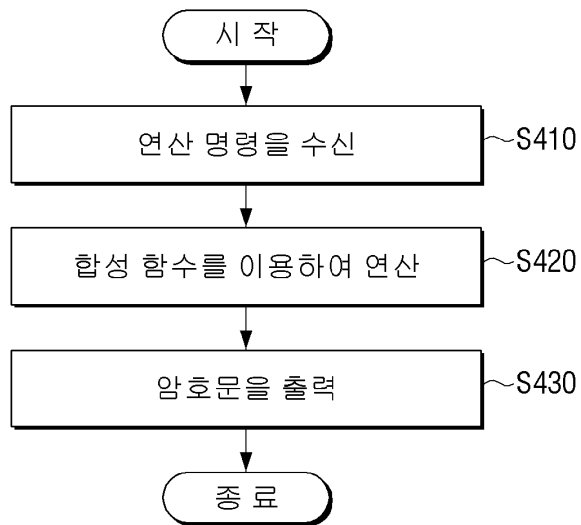
[도2]



[도3]



[도4]



[도5]

Algorithm 1 Inv(x;d)

Input: $0 < x < 2, d \in \mathbb{N}$ Output: an approximate value of $1/x$ (refer Lemma 1)

```

1:  $a_0 \leftarrow 2 - x$ 
2:  $b_0 \leftarrow 1 - x$ 
3: for  $n \leftarrow 0$  to  $d - 1$  do
4:    $b_{n+1} \leftarrow b_n^2$ 
5:    $a_{n+1} \leftarrow a_n \cdot (1 + b_{n+1})$ 
6: end for
7: return  $a_d$ 
  
```

[도6]

Algorithm 2 Sqrt(x;d)

Input: $0 \leq x \leq 1, d \in \mathbb{N}$ Output: an approximate value of $\sqrt{x}$ (refer Lemma 2)

```

1:  $a_0 \leftarrow x$ 
2:  $b_0 \leftarrow x - 1$ 
3: for  $n \leftarrow 0$  to  $d - 1$  do
4:    $a_{n+1} \leftarrow a_n \left(1 - \frac{b_n}{2}\right)$ 
5:    $b_{n+1} \leftarrow b_n^2 \left(\frac{b_n - 3}{4}\right)$ 
6: end for
7: return  $a_d$ 
  
```

[57]

Algorithm 3 Min(a, b; d), Max(a, b; d)

Input: $a, b \in [0, 1]$, $d \in \mathbb{N}$

Output: an approximate value of min(a, b) and max(a, b) (refer Theorem 1,2)

$$1: x = \frac{a+b}{2} \text{ and } y = \frac{a-b}{2}$$

$$2: z \leftarrow \text{Sqrt}(y^2; d)$$

$$3: \text{return } \begin{array}{l} x - z \text{ for Min}(a, b; d) \\ x + z \text{ for Max}(a, b; d) \end{array}$$

[58]

Algorithm 4 Comp(a, b; d, d', t, m)

Input: distinct numbers $a, b \in \left[\frac{1}{2}, \frac{3}{2}\right)$, $d, d', t, m \in \mathbb{N}$

Output: an approximate value of comp(a, b) (refer Theorem 4)

$$1: a_0 \leftarrow \frac{a}{2} \cdot \text{Inv}\left(\frac{a+b}{2}; d'\right)$$

$$2: b_0 \leftarrow 1 - a_0$$

3: for $n \leftarrow 0$ to $t - 1$ do

$$4: \text{inv} \leftarrow \text{Inv}(a_n^m + b_n^m; d)$$

$$5: a_{n+1} \leftarrow a_n^m \cdot \text{inv}$$

$$6: b_{n+1} \leftarrow 1 - a_{n+1}$$

7: end for

8: return a_t

[59]

Algorithm 5 NewComp(a, b; n, d)

Input: $a, b \in [0, 1]$, $n, d \in \mathbb{N}$ Output: An approximate value of 1 if $a > b$, 0 if $a < b$ and 1/2 otherwise

$$1: x \leftarrow a - b$$

2: for $i \leftarrow 1$ to d do

$$3: x \leftarrow f_n(x)$$

// compute $f_n^{(d)}(a - b)$ for input a and b

4: end for

$$5: \text{return } (x+1)/2$$

[510]

Algorithm 6 FindG(n, c)Input: $n \in \mathbb{N}$, $0 < c < 1$ Output: A degree- $(2n+1)$ polynomial $g_{n,c}$ satisfying Prop I & IV with minimal δ of Prop IV.

```

1:  $g_{n,c} \leftarrow x$ 
2: repeat
3:  $\delta_0 \leftarrow$  minimal  $\delta$  such that  $g_{n,c}([\delta, 1]) \subseteq [1-c, 1]$ 
4:  $g_{\min} \leftarrow$  degree- $(2n+1)$  minimax approx. poly. of  $(1 - \frac{c}{2}) \cdot \text{sgn}(x)$  over  $[-1, -\delta_0] \cup [\delta_0, 1]$ 
5:  $g_{n,c} \leftarrow g_{\min}$ 
6:  $S \leftarrow \|g_{n,c} - (1 - \frac{c}{2})\|_{\infty, [\delta_0, 1]}$ 
7: unit  $S = \frac{c}{2}$ 
8: return  $g_{n,c}$ 

```

[511]

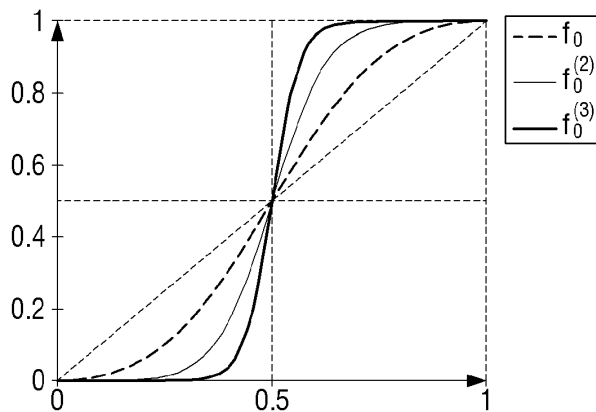
Algorithm 7 NewCompG(a, b; n, d_f , d_g)Input: $a, b \in [0, 1]$, $n, d_f, d_g \in \mathbb{N}$ Output: An approximate value of 1 if $a > b$, 0 if $a < b$ and 1/2 otherwise

```

1:  $x \leftarrow a - b$ 
2: for  $i \leftarrow 1$  to  $d_g$  do
3:  $x \leftarrow g_n(x)$  // compute  $g_n^{(d_g)}(a - b)$  for input a and b
4: end for
5: for  $i \leftarrow 1$  to  $d_f$  do
6:  $x \leftarrow f_n(x)$  // compute  $f_n^{(d_f)}(a - b)$  for input a and b
7: end for
8: return  $(x+1)/2$ 

```

[512]



[도 13]

