



최대공약수 알고리즘

유클리드 호제법

> 최대공약수란?

- 최대공약수 :

공약수란 두 수 이상의 여러 수의 공통된 약수를 의미한다.

즉, 최대공약수란 두 수 이상의 여러 수의 공약수 중 최대인 수를 말한다.

EX) 72와 90의 최대 공약수

$$72 = 2*2*2*3*3$$

$$90 = 2*3*3*5$$

$$= 2*3*3$$

> 유클리드 호제법이란?

- 유클리드 호제법 :

두 수가 서로 상대방 수를 나누어서 2개의 자연수의

최대공약수를 구하는 알고리즘이다.

$a > b$ 일때 2개의 자연수 a, b 에 대해서 a 를 b 로 나눈 나머지를 r 이라 하면

a 와 b 의 최대공약수는 b 와 r 의 최대공약수이다.

즉, $a > b$ 일때, $a \% b = r$ 이며, a 와 b 의 최대공약수를 $\text{gcd}(a,b)$ 라고 하면,

$$\text{gcd}(a,b) = \text{gcd}(b,r)$$

> 유클리드 호제법이란?

- 유클리드 호제법 : 계산 방법 예시 1071과 1029의 최대공약수 구하기

1071은 1029로 나누어 떨어지지 않기 때문에

1071을 1029로 나눈 나머지를 구한다 \gg 42

1029는 42로 나누어 떨어지지 않기 때문에

1029를 42로 나눈 나머지를 구한다 \gg 21

42는 21로 나누어 떨어진다.

q따라서, 최대공약수는 21이다.

> 유클리드 호제법이란?

- 유클리드 호제법 : 장점과 단점

장점 -

가장 단순한 방법인 모든 정수로 나누어 보는 방법보다
시간 복잡도를 줄일 수 있어서 좀 더 효율적이고 빠른 알고리즘 작성이 가능하다.

단점 -

최대공약수는 빠르게 산출이 가능하지만 최소공배수를 계산함에 있어서는
비교 대상의 초기값 a 와 b 를 특정 변수에 저장을 시키고 있어야 해서 비효율적이다.

> 유클리드 호제법이란?

- 유클리드 호제법이 아닌 브루트 포스 :

```
int GCD () {  
    int a = 9, b = 15;  
    int min_num = (a < b ? a : b);  
  
    for (int i=min_num; i>0; i--) {  
        if (a % i == 0 && b % i == 0) {  
            cout << i;  
            break;  
        }  
    }  
}
```

> 유클리드 호제법이란?

- 유클리드 호제법 : 재귀 함수

```
int GCD(int a, int b) {  
    if(b==0) return a;  
    else return GCD(b,a%b);  
}
```

> 유클리드 호제법이란?

- 유클리드 호제법 : 반복문 $a > b$ 일때

```
int GCD (int a, int b) {  
    while(1){  
        int r = a%b;  
        if(r==0) return b;  
        a = b; b = r;  
    }  
}
```




끝!