수학 2 (참고)

최백준 choi@startlink.io

피보나치

피보나치 수

Fibonacci Number

- $\sum_{i=1}^{n} F_i = F_{n+2} - 1$
- $\sum_{i=1}^{n} F_{2i} = F_{2n+1} - 1$
- $\sum_{i=0}^{n} F_{2i+1} = F_{2n}$
- $\sum_{i=1}^{n} F_i^2 = F_n F_{n+1}$
- $\gcd(F_n, F_m) = F_{\gcd(n,m)}$

그 외의 피보나치 수 문제

Fibonacci Number

- 피보나치 수 4: https://www.acmicpc.net/problem/10826
- 피보나치 수 5: https://www.acmicpc.net/problem/10870
- 피보나치 수의 확장: https://www.acmicpc.net/problem/1788
- 피사노 주기: https://www.acmicpc.net/problem/9471
- 피보나치 수의 합: https://www.acmicpc.net/problem/2086
- 피보나치 수의 제곱의 합: https://www.acmicpc.net/problem/11440
- 홀수번째 피보나치 수의 합: https://www.acmicpc.net/problem/11442
- 짝수번째 피보나치 수의 합: https://www.acmicpc.net/problem/11443
- 피보나치 수와 최대공약수: https://www.acmicpc.net/problem/11778

이항 계수

뤼카의 정리

Lucas' Theorem

- 음이 아닌 정수 n, m과 소수 p에 대해서 다음이 성립한다.
- $\binom{n}{m} = \prod_{i=0}^{k} \binom{n_i}{m_i} \pmod p$
- 여기서 n_i와 m_i는 n과 m을 p진법으로 나타낸 것이다. 즉, 다음과 같다.
- $n = n_k p^k + n_{k-1} p^{k-1} + \cdots + n_1 p + n_0$
- $m = m_k p^k + m_{k-1} p^{k-1} + \cdots + m_1 p + m_0$

이항 계수 4

https://www.acmicpc.net/problem/11402

- $\binom{n}{k}$ mod M을 구하는 문제
- $1 \leq n \leq 10^{18}$, $0 \leq k \leq n$, $2 \leq m \leq 2000$, m은 소수
- m이 2000보다 작은 소수이기 때문에, 파스칼의 삼각형을 만들고
- 뤼카의 정리를 이용하면 된다.

이항 계수 4

https://www.acmicpc.net/problem/11402

- 소스: http://codeplus.codes/122e536bea3449c2ae04b341035652b3

이항 계수 5

https://www.acmicpc.net/problem/11439

- $\binom{n}{k}$ mod M을 구하는 문제
- $1 \le n \le 4 \times 10^6$, $0 \le k \le n$, $2 \le m \le 4 \times 10^6$, m은 소수
- $\binom{n}{k}$를 소인수 분해 하면서 풀어야 한다.
- 팩토리얼 0의 개수 문제를 풀 때, N!를 소인수 분해를 하면 5^k의 k가 몇 개 인지 구하는 방법을 배웠다.
- 이 방법을 응용해서 푼다.

이항 계수 5

https://www.acmicpc.net/problem/11439

- 소스: http://codeplus.codes/c6570bc13e074b6c9d4195d9235f4883

오일러 피 함수

오일러 피 함수

Euler's phi Function

- $\phi(n)$로 나타낸다.
- $\phi(n)$ = gcd(n, k) = 1 인 $1 \leq k \leq n$의 개수
- $\phi(9) = 6$
- k = 1, 2, 4, 5, 7, 8

오일러 피 함수

Euler's phi Function

- ϕ(p) = p-1 (p가 소수인 경우)
- ϕ(nm) = ϕ(n) × ϕ(m) (n, m이 서로소인 경우)
- ϕ(n) = $n \prod_{p \mid n}(1 - \frac{1}{p})$
- p는 n의 소인수
- ϕ(9) = 9 * (1 – 1/3) = 9 * 2/3 = 6

오일러 피 함수

Euler's phi Function

```
long long phi(long long n) {
    long long ans = n;
    for (long long i=2; i*i<=n; i++) {
        if (n % i == 0) {
            while (n % i == 0)
                n /= i;
            ans -= ans / i;
        }
    }
    if (n > 1)
        ans -= ans / n;
    return ans;
}
```

GCD(n,k) = 1

https://www.acmicpc.net/problem/11689

- 오일러 피 함수를 구현해보는 문제

GCD(n,k) = 1

https://www.acmicpc.net/problem/11689

- 소스: http://codeplus.codes/e9d40749f137465498ee0cf1366434cc

확장 유클리드 알고리즘

확장 유클리드 알고리즘

Extended Euclidean Algorithm

- $ax + by = gcd(a,b)$ 의 해를 구할 수 있는 알고리즘
- 유클리드 알고리즘과 다르게 4개의 변수를 이용해서 사용한다.
- gcd(a, b)를 조금 어렵게 써보면 다음과 같다.
- 몫: $q_0, \cdots, q_k$, 나머지: $r_0 \cdots, r_k$
- $r_0 = a$
- $r_1 = b$
- …
- $r_{i+1} = r_{i-1} - q_i r_i$ ($0 \leq r_{i+1} < |r_i|$, 이 부등식으로 q_i를 결정할 수 있다)

확장 유클리드 알고리즘

Extended Euclidean Algorithm

- 유클리드 알고리즘에 두 변수 s와 t를 추가해야 한다.
- $r_0 = a, r_1 = b$
- $s_0 = 1, s_1 = 0$
- $t_0 = 0, t_1 = 1$
- …
- $r_{i+1} = r_{i-1} - q_i r_i \ (0 \leq r_{i+1} < |r_i|)$
- $s_{i+1} = s_{i-1} - q_i s_i$
- $t_{i+1} = t_{i-1} - q_i t_i$

확장 유클리드 알고리즘

Extended Euclidean Algorithm

- a = 240, b= 46인 경우를 풀어보자.
- 240x + 46y = gcd(240, 46) = 2

i	q_{i-1}	r_i	s_i	t_i
0		240	1	0
1		46	0	1
2	240/46 = 5	240 - 5 ×46 = 10	1 - 5×0 = 1	0 - 5×1 = -5
3	46/10 = 4	46 - 4×10 = 6	0 - 4×1 = -4	1 – 4×-5 = 21
4	10/6 = 1	10 - 1×6 = 4	1 - 1×-4 = 5	-5 - 1×21 = -26
5	6/4 = 1	6 - 1×4 = 2	-4 - -1×5 = -9	21 - 1×-26 = 47
6	4 × 2 = 2	4 - 2×2 = 0	5 – 2×-9 = 23	-26 - 2×47 = -120

확장 유클리드 알고리즘

Extended Euclidean Algorithm

- $ax + by = gcd(a,b)$ 의 해를 쉽게 구할 수 있는 알고리즘
- 대부분의 경우에 a와 b중 하나는 음수가 나온다.
- $240x + 46y = gcd(240, 46) = 2$
- 확장 유클리드 알고리즘의 마지막 이전 s와 t값이 x와 y값이 된다.
- $240 \times -9 + 46 \times 47 = 2$

확장 유클리드 알고리즘

Extended Euclidean Algorithm

- $ax + by = gcd(a,b)$ 의 해를 쉽게 구할 수 있는 알고리즘
- 이 알고리즘은 gcd(a,b)가 1인 경우에 유용하게 사용할 수 있다.
- $ax + by = 1$ 일 때, x는 a의 나머지 연산의 곱셈 역원이 되기 때문

나머지 연산의 곱셈 역원

Modular Multiplicate Inverse

- 정수 a을 m으로 나눈 나머지의 곱셈 역원은 $a \times a^{-1} \equiv 1 \pmod{m}$ 을 만족하는 a^{-1} 을 말한다.
- 즉, $a^{-1} \equiv x \pmod{m}$ 을 만족하는 x를 말한다.
- 역원은 a와 m이 서로소인 경우에만 존재한다.

```
for (int i=1; i<m; i++) {
    if ((a*i) % m == 1) {
        x = i;
    }
}
```

- 위 소스의 시간 복잡도는 O(m) 이다.

나머지 연산의 곱셈 역원

Modular Multiplicate Inverse

- 확장 유클리드 알고리즘을 이용해서 구할 수도 있다.
- ax = 1 (mod m) 을 구해야 하기 때문에
- ax = 1 + my로 바꿔서 쓸 수 있다.
- ax – my = 1 로 다시 쓸 수 있고, x와 y는 음수가 되어도 상관 없기 때문에
- ax + my = 1 로 다시 바꿔 쓸 수 있다.
- 이제 확장 유클리드 알고리즘을 이용해서 x의 값을 구할 수 있다.

나머지 연산의 곱셈 역원

Modular Multiplicate Inverse

- m이 소수인 경우에는 페르마의 소정리를 이용해서 구할 수도 있다.
- m이 소수이고, a가 m과 서로소라면, a^{m-1}은 m으로 나눈 나머지는 1이다.
- 즉
- $a^{m-1} \equiv 1 \pmod{m}$ 이라는 의미이다.
- 따라서, $a \times a^{m-2} \equiv 1 \pmod{m}$ 이고,
- a^{m-2} 가 $a \times x \equiv 1 \pmod{m}$ 을 만족하는 x가 되기 때문에
- 역원은 a^{m-2}가 된다.

이항 계수 3

https://www.acmicpc.net/problem/11401

- $\binom{n}{k}$ mod M을 구하는 문제
- $1 \leq n \leq 4,000,000$, $0 \leq k \leq n$, $M = 1,000,000,007$
- 나머지 연산의 곱셈 역원을 이용해서 풀 수 있다.

이항 계수 3

https://www.acmicpc.net/problem/11401

- 소스: http://codeplus.codes/93d563fc0ba944f6ae728535fc6ecbac