

#고급반 3주차

고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

^{2p} / <정수론>

^{22p} / <조합론>

^{34p} / <연습 문제>

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

정수론 (Number Theory) – 정수를 다루는 수학의 한 분야

1. 정수론 개념은 고난이도 문제에서 자주 등장하는 만큼 고급반 이상인 분들에게는 필수적인 분야
2. 많은 경우 개념을 사전지식으로 알고 있어야 하기에 모르면 접근조차 어려워지는 경우도 상당
3. <https://rkm0959.tistory.com/> 에게 모든 영광을 바칩니다

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

페르마 소정리 (Fermat's little Theorem) – 여백이 부족해 설명은 밑에 적습니다

1. p 가 소수일때, 모든 정수 a 에 대해 $a^p \equiv a \pmod{p}$
2. p 가 소수이고 a 가 p 의 배수가 아니면
 $a^{p-1} \equiv 1 \pmod{p}$

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

이항정리와 귀납법을 사용하여 간단하게
증명해봅시다.

우선 페르마 소정리는 다음과 동치이며
“ p 가 소수라면 $n^p \equiv n \pmod{p}$ ” 이를
귀납법을 위해 정수 n 에 대해 참인 명제라고 가
정합니다.

이항정리에 의해 $(n+1)^p = n^p + 1 + \sum_{i=1}^{p-1} \binom{p}{i} n^i$

이때 $\binom{p}{i} = \frac{p(p-1)\dots(p-i+1)}{i!}$ 이며 이는 자명하게
 p 의 배수이다.

따라서, $(n+1)^p \equiv n^p + 1 \pmod{p}$ 이며 앞서
참이라 가정한 명제로 인해

$$(n+1)^p \equiv n+1 \pmod{p}$$

가 성립하게 된다. $(n-1)$ 때도 마찬가지로
성립하므로 모든 정수 n 에 대해 공식이
성립함을 귀납적으로 증명할 수 있다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

모듈러 곱셈 역원 (Modular Multiplicative Inverse) – $ax \equiv 1(\text{mod } N)$ 을 만족하는 정수 x 를 a 의 $\text{mod } N$ 에 대한 역원이라고 부릅니다.

1. 진작에 페르마 소정리를 배운이유, 모듈러 사칙연산에서 나누기에 해당하는 부분을 곱셈으로 바꿀 수 있게 된다
2. a 와 N 이 서로소가 아니라면 역원은 존재하지 않는다

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

페르마 소정리를 사용하여 N 이 소수인 경우에
모듈러 곱셈 역원을 어떻게 구할 수 있는지
봅시다.

페르마 소정리를 다시 기억해봅시다. p 가 소수
이고 a 가 p 의 배수가 아니면 $a^{p-1} \equiv 1 \pmod{p}$

$$a^{p-1} = a \cdot a^{p-2}$$

$$a \cdot a^{p-2} \equiv 1 \pmod{p}$$

따라서 a 의 p 에 대한 역원은 a 의 $p-2$ 제곱임을
알 수 있습니다.

예시)

$$n=2, p=7$$

$$2^{7-2} = 32$$

$$2(32) = 64$$

$$64 \equiv 1 \pmod{7}$$

정리가 성립함을 볼 수 있다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

시간 복잡도 - $O(\log(p))$

1. 결론적으로 계산하는 것은 a 의 $p-2$ 제곱
2. 이는 분할정복을 통한 거듭제곱으로 \log 시간에 처리 가능

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

4 Σ (BOJ #13172)

<문제 설명>

- 페르마 소정리 와 모듈러 곱셈 역원에 대한 설명이
길게 나와 있는 문제
- 기약분수/나눗셈을 모듈러 사칙연산에서 처리 할 방
법이 생겼음이 중요
- “어떤 분수가 기약분수로 나타냈을 때 a/b 이면, 이 분
수는 $a \cdot b^{-1} \bmod X$ (X 는 소수)으로 대신 계산하도록
한다. 여기서 b^{-1} 은 b 의 모듈러 곱셈에 대한 역원이다.”

<제약 조건>

- $1 \leq M \leq 10,000$
- $1 \leq N_i, S_i \leq 1,000,000,000$

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

확장 유클리드 (Extended Euclidean) – 정수 a, b 가 주어질 때 $ax + by = \gcd(a, b)$ 를 만족하는 정수쌍 (x, y) 를 찾는 알고리즘

1. 기존의 유클리드 알고리즘은 최대공약수만을 찾아줬지만 확장 유클리드에서는 위 식을 만족하는 정수 쌍 (x, y) 까지 찾아줍니다
2. 추가적으로 모듈러 곱셈 역원을 a 와 p 가 서로소인 모든 경우에 대해 구할 수 도 있습니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

확장 유클리드 알고리즘을 다루기 전에
유클리드 알고리즘에 대해 다시 한번 자세하게
살펴보도록 합시다.

앞으로 나올 문단들에서는 어떠한 수 b 가 a 로
나누어 떨어지면 (나머지가 0) $a \mid b$ 라고 표현
하겠습니다.

우선, 유클리드 알고리즘은 두 자연수 a, b 가
주어졌을 때 이 두 수의 최대공약수를 구하는 알고리
즘입니다.

1. 몫, 나머지 (q, r)를 사용해 a 를 b 에 대해 표현하면
 $a = bq + r$

$\gcd(a, b)$ 를 g 라고 할때, $g \mid a$ 와 $g \mid b$ 는 자명하게 성
립합니다. 따라서 $g \mid (a - bq)$, $a - bq = r$
 $g \mid b, g \mid r$

역으로 이번에는 $\gcd(b, r)$ 이 g 인 경우를 생각해보면
 $a = bq + r$ 이므로 $g \mid a$
 $g \mid a, g \mid b$

따라서 “ g 가 a, b 의 공약수이다”와 “ g 가 b, r 의 공약수
이다”는 완전히 동치가 됩니다. 따라서 우리는

$\gcd(a, b) = \gcd(b, r)$ 이라는 결론에
도달할 수 있습니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

2. 직전 슬라이드에서의 내용은 a, b 에 대한 문제를 b, r 이라는 더 작은 수 범위에 대한 문제로 줄일 수 있게 해줍니다.

여기서 증명은 하지 않지만 $br \leq (1/2) ab$ 라는 부등식을 유도할 수 있으며 이는 각 단계에서 범위가 절반 이상 감소함을 보여주며 이것이 바로 유클리드 알고리즘이 $\log(\max(a, b))$ 시간에 작동한다는 근거가 됩니다.

작동 예시)

$$\gcd(21, 10) = ?$$

$$21 = 2(10) + 1$$

$$\gcd(10, 1) = ?$$

$$10 = 1(10) + 0$$

$$\gcd(1, 0) = 1$$

(일반적으로 재귀 함수의

종료 조건을 정의 하기 위해 $\gcd(n, 0) = n$ 으로 정의합니다)

$$\text{따라서 } \gcd(21, 10) = \gcd(1, 0) = 1$$

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

이제 본격적으로 확장 유클리드 알고리즘을 다뤄보도록 합시다.

일단 지금은 $\gcd(a, b) = 1$ 임을 가정합시다.
이러면 문제는 $ax + by = 1$ 을 푸는 문제로 변환됩니다.

이번에도 마찬가지로 $a = bq + r$ 의 형태로 나타내봅시다.

$$ax + by = 1$$

$$(bq + r)x + by = 1$$

$$b(qx + y) + rx = 1$$

$$x' = qx + y, y' = x$$

$$bx' + ry' = 1$$

위 과정을 통해 우리는 **유클리드 알고리즘의 동작과 동일하게 (a, b) 에 대한 문제를 (b, r) 에 대한 문제로 축소할 수 있습니다.** 마찬가지로 시간 복잡도는 로그 시간.

위 과정을 재귀적으로 $(1, 0)$ 에 대한 문제로 축소하면 이는 $x = 1, y = 0$ 이라는 해를 가지게 되며 $x = y', y = x' - qx$ 라는 점을 통해 (x, y) 또한 빠르게 찾을 수 있습니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

마찬가지로 $ax + by = n$ 이라는 식을 해결하는 문제를 봅시다. $\gcd(a, b)$ 는 유클리드 알고리즘으로 구할 수 있으며, $g \mid (ax + by)$ 는 자명하게 성립합니다.

따라서 만약 n 이 g 의 배수가 아니라면 해가 존재하지 않습니다.

만약 $g \mid n$ 이라면 좌변 우변을 g 로 나눕니다
 $(a/g)x + (b/g)y = n/g$
 $\gcd((a/g), (b/g)) = 1$ (g 가 \gcd 라서)

따라서 이 문제를 $(a/g)x' + (b/g)y' = 1$ 을 해결한 후에 그 값들로 부터 x, y 를 다시 계산하는 문제로 바꿀수 있으며 이는 이전 슬라이드와 동일하게 해결 가능합니다.

추가적으로, 정수쌍 (x, y) 를 해로 얻었다면 우리는 이 식의 일반해 또한 정수 t 에 대해 $(x + (b/g)t, y - (a/g)t)$ 로 나타낼수 있습니다.

또한 지금은 유클리드 알고리즘을 통해 \gcd 를 먼저 구하라고 했지만 유클리드와 확장 유클리드 모두 범위를 축소하는 방식이 똑같다는 점을 떠올리면, 한번의 실행으로 \gcd 까지 구해줄수 있습니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

잠깐! 분명 이거로 모듈러 곱셈 역원 까지 구할 수 있다고 했잖아요.

네 그렇습니다. 곱셈 역원을 구하는 문제를 $ax \equiv 1 \pmod{N}$ 을 만족하는 x 를 구하는 문제로 생각하면.

$$ax \equiv 1 \pmod{N}$$

$$ax + Ny = 1$$

즉 확장 유클리드 알고리즘을 통해 위 식을 만족하는 (x, y) 를 찾으면 x 가 바로 곱셈 역원입니다. 오른쪽은 제 구현입니다.

```
//typedef tuple<ll, ll, ll> t13;

t13 egcd(ll a, ll b) {

    if(b==0) return {1LL, 0LL, a};
    t13 res = egcd(b, a%b);

    t13 ret = {get<1>(res), get<0>(res) -
a/b*get<1>(res), get<2>(res)};

    return ret;
}
```

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

2 역원(Inverse) 구하기 (BOJ #14565)

<문제 설명>

- 확장 유클리드를 통해 곱셈 역원을 구하는 문제. 덧셈 역원은 흠...
- **오버플로우**에 주의하고 **확장 유클리드 알고리즘의 해가 없는 조건**이 무엇인지 기억해보시길 바랍니다.

<제약 조건>

- $2 \leq N \leq 10^{12}$
- $1 \leq A < N$

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

중국인의 나머지 정리 (Chinese Remainder Theorem) – 손자(孫子)가 물었다,
3으로 나누었을 때 2가 남고, 5로 나누었을 때 3이 남고, 7로 나누었을 때 2가 남는 수는 무엇인가?

1. 중국인의 나머지 정리 (이하 CRT/중나정) 은 위와 같은 연립 합동식을 해결하기 위한 알고리즘 입니다.
2. 여기까지만 해도 웬만한 PS/CP문제를 해결함에 있어 정수론 사전지식은 충분하다는 의견이 대중적입니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

합동식 - $ax \equiv b \pmod{c}$ 등 모듈러 연산을 활용하는 방정식

1. 연립 합동식 또한 결국 모듈러 연산을 활용하는 방정식들의 연립방정식
2. 즉, 아까 우리가 풀던 문제중에는 합동식도 있었습니다. 새로운 용어에 겁먹지 마시길 바랍니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

$$x \equiv a_1 \pmod{n_1}$$

$$x \equiv a_2 \pmod{n_2}$$

위 두 식이 성립한다고 가정하면 정의상

$x = n_1 y + a_1$ 인 정수 y 가 존재

$$n_1 y + a_1 \equiv a_2 \pmod{n_2}$$

$$n_1 y \equiv a_2 - a_1 \pmod{n_2}$$

즉 위 식을 해결하면 됩니다.

그리고 이 식은 결국 $n_1 y + n_2 x = a_2 - a_1$

즉, 확장 유클리드 알고리즘으로 해결할수 있는

식이 되버립니다.

앞서 봤듯이 이번에도 $a_2 - a_1$ 이 $\gcd(n_1, n_2)$ 의 배수가 아니라면 해가 없습니다.

우선 한가지 알아야 할 것 한가지는

합동식의 해는

$x \equiv (\text{something}) \pmod{(\text{something})}$ 의 형태로

주어진다는 점. 저 조건만 만족하는 수는 모두

해라는 의미이며 그러한 수는 무한히 많다.

그리고 지금의 경우 만약 해가 있다면 약간

재밌는 결과를 얻을수 있는데 바로 해는 무조건

$x \equiv (\text{something}) \pmod{\text{lcm}(n_1, n_2)}$ 의 형태

라는 점입니다. (lcm 은 최소공배수)

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

해가 존재한다는 가정하에 $g = \gcd(n_1, n_2)$ 라고
합시다.

$$n_1 y \equiv a_2 - a_1 \pmod{n_2}$$

양변을 g 로 나누면

$$(n_1/g) y \equiv (a_2 - a_1)/g \pmod{n_2/g}$$

$$y \equiv (a_2 - a_1)/g \cdot (n_1/g)^{-1} \pmod{n_2/g}$$

$$(a_2 - a_1)/g \cdot (n_1/g)^{-1} = t \text{ (임시 변수 } t)$$

$$y \equiv t \pmod{n_2/g}$$

$$(n_2/g) m + y = t$$

$$y = t - (n_2/g) m$$

$x = n_1 y + a_1$ 에 대입하면

$$x = n_1(t - (n_2/g) m) + a_1$$

$$x = n_1 t - (n_1 n_2/g) m + a_1$$

$$x + (n_1 n_2/g) m = n_1 t + a_1$$

$$x \equiv n_1 t + a_1 \pmod{n_1 n_2/g}$$

여기서 $n_1 n_2/g = \text{lcm}(n_1, n_2)$

따라서 $x \equiv n_1 t + a_1 \pmod{\text{lcm}(n_1, n_2)}$

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

즉, 두 합동식을 하나의 합동식으로 합쳐나가는 과정을 모든 합동식이 하나로 합쳐질때 까지 반복해주면 됩니다. 그리고 그때 해는 없거나 something (mod $\text{lcm}(n_1, n_2)$) 의 형태입니다.

해가 없는 경우가 발생하면 전체 연립 합동식 또한 해가 없음을 의미합니다.

오른쪽은 제 구현입니다.

```
t13 egcd(ll a, ll b) {
    if(b==0) return {1LL, 0LL, a};
    t13 res = egcd(b, a%b);
    t13 ret = {get<1>(res), get<0>(res)-a/b*get<1>(res),
get<2>(res)};
    return ret;
}

p11 crt(ll a1, ll n1, ll a2, ll n2) {
    t13 tmp = egcd(n1, n2);
    // egcd == 확장 유클리드

    ll g = get<2>(tmp);
    ll lcm = n1/g*n2;

    if((a2 - a1)%g!=0) {
        return {-1, -1};
    }
    // 리턴이 -1, -1인 경우 해가 없음을 확인

    ll t = ((a2-a1)/g)%(n2/g) * (get<0>(egcd(n1/g,
n2/g)))%(n2/g);

    ll ret = n1*(t%(n2/g))+a1;
    ret = (ret%lcm+lcm)%lcm;
    // return {something, lcm}
    return {ret, lcm};
}
```

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

4 백남이의 여행 준비의 준비 (BOJ #23062)

<문제 설명>

- 대놓고 중나정을 사용하라고 존재하는 문제
- 역시나 오버플로우에 주의하고 이를 해결하기 위한 처리에 대해 고민해보면 좋습니다

<제약 조건>

- $1 \leq T \leq 1,000,000$
- $1 \leq A, B, C \leq 1,000,000$
- $0 \leq a < A$
- $0 \leq b < B$
- $0 \leq c < C$

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

조합론 (Combinatorics) – 여러 원소의 조합과 순열을 다루는 수학의 한 분야

1. 조합론 또한 다양한 난이도 분포에서 자주 등장하며
개인차에 따라 체감 난이도를 크게 타는 분야중 하나
2. 마찬가지로 많은 경우 개념을 사전지식으로 알고 있어야 하기에 모르면 접근조차 어려워지는 경우도 상당
3. 개념만으로 풀리는 문제보다는 답을 구하는 과정의 일부로 개념을 적용하는 경우가 다수
4. 3으로 인해 실질적으로 코드로 구현해봐야만 하는것은 한 부분밖에 없음

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

포함 배제의 원리 (Inclusion-Exclusion Principle) – 조합론에서 여러 개의 합집합의 크기를 구할 때 사용하는 공식

The Inclusion-Exclusion Principle is a really powerful math concept.

It starts out with a grade school level observation and builds up to this truly unhinged looking equation.

It seems scary but read this thread and you'll be one of the happy few that understands it.

[트윗 번역하기](#)

INCLUSION-EXCLUSION PRINCIPLE

$$\begin{aligned}
 |A_1 \cup \dots \cup A_n| &= \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| \\
 &\quad + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \dots \\
 &\quad + (-1)^{n+1} |A_1 \cap \dots \cap A_n|
 \end{aligned}$$

 @kareem_carr

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

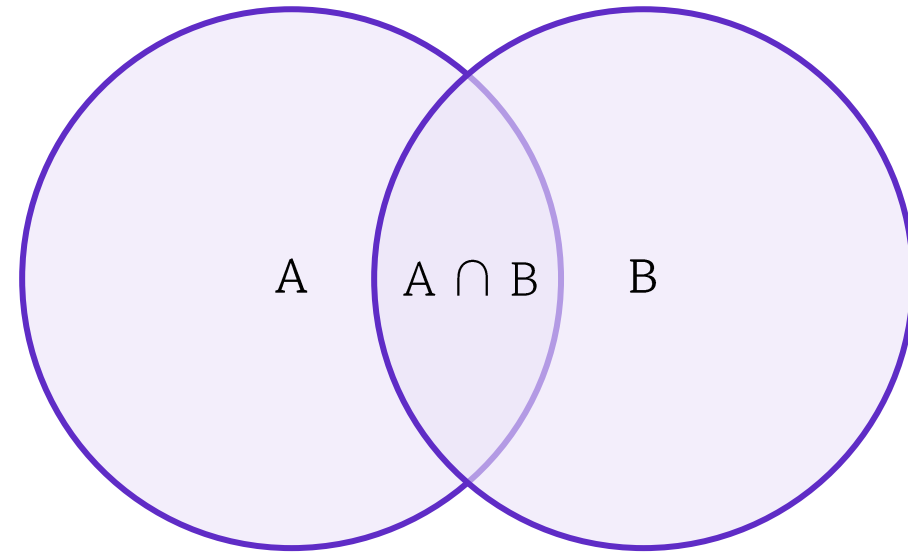
증명은 쉽지 않지만 시각적으로 이해하기는
어렵지 않으니 최대한 직관적으로 이해해보는
방향으로 설명을 해보겠습니다.

우선 두개의 집합이 있는 경우부터 오른쪽과
같이 상상해보겠습니다. 어떠한 집합 S 의 크기를 $|S|$
라 하면

$$|A \cup B| = |A| + |B| - |A \cap B|$$

이 성립합니다. 여기서 $(A \cup B)$ 는 A 와 B 의
합집합. $A \cap B$ 는 A 와 B 의 교집합입니다.

시각적으로는 A 의 크기와 B 의 크기를 더했을때 양쪽
에 속하는 교집합의 크기가 두번 더해졌기에 한번을
빼준다고 이해해볼수 있습니다.



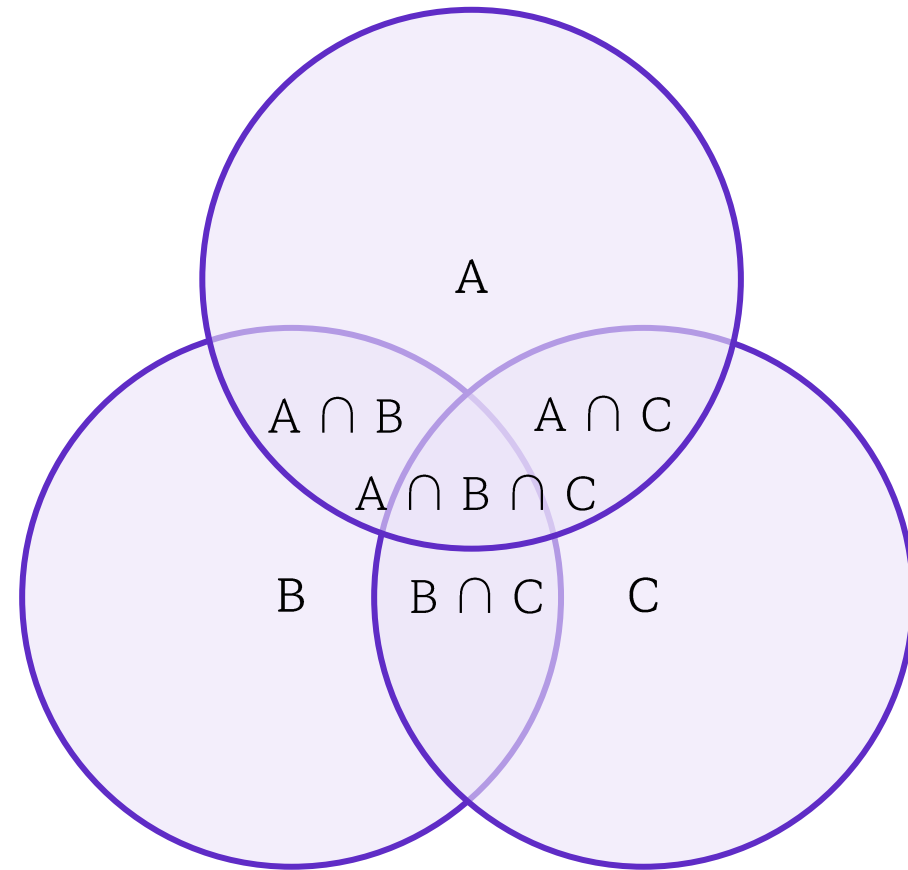
#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

그러면 그 다음은 집합 3개를 가지고 똑같이 식을 만들어 봅시다.

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$$


A, B, C의 크기를 더하고 두개씩 겹친거 ($A \cap B$, $B \cap C$, $A \cap C$)를 한번씩 빼주는 것까지는 동일하지만 이번엔 **두개씩 겹친거를 세 번 빼주면서 A, B, C가 모두 겹친 $A \cap B \cap C$ 가 한번 더 빼졌기에** 이를 한번 다시 더해준겁니다.



#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

이러한 패턴은 집합의 개수가 늘어나면서
반복되고 집합 n개에 대해서 일반화한 공식은
오른쪽과 같습니다.

Kareem Carr | Data Scientist 
@kareem_carr

The pattern continues for more and more sets.

So the union of n sets looks like this:

트윗 번역하기

The size of the union of n sets


The size of each set individually

The size of the intersection of all combinations of two sets

The size of the intersection of all combinations of three sets

The size of the intersection of all n sets

$$|A_1 \cup \dots \cup A_n| = \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \dots + (-1)^{n+1} |A_1 \cap \dots \cap A_n|$$

 @kareem_carr

출처: https://twitter.com/kareem_carr/status/1632061114069704704?s=20

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

비둘기집 원리 (Pigeonhole's Principle) – $n + 1$ 개의 비둘기가 n 개의 비둘기 집에 들어갈 경우, 최소한 하나의 비둘기집에는 반드시 비둘기가 두마리 이상 들어가있다

1. 생각보다 직관적으로 이해 가능하고 강력한 원리입니다. 증명도 귀류법으로 간단히 가능하나 여기서는 하지 않겠습니다.
2. 이 원리의 확장판으로 $kn + 1$ 마리의 비둘기가 있다면 최소한 하나의 비둘기집에는 $k + 1$ 마리의 비둘기가 들어갑니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

기댓값의 선형성 (Linearity of Expectations) – $E(X + Y) = E(X) + E(Y)$

1. 두 이산 확률 변수 X 와 Y 가 같은 확률 공간 S 에 있다고 했을때
2. $E(X)$ 가 X 의 기댓값이고 $E(Y)$ 가 하면
 $E(X + Y) = E(X) + E(Y)$
3. k 개의 확률 변수에 대해 다음과 같이 일반화 가능
 $E[\sum_{i=1}^k c_i X_i] = \sum_{i=1}^k c_i E[X_i]$ (여기서 c 는 임의의 계수)
4. 두 확률변수가 독립이 아니여도 성립

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

간단하게 증명을 해보도록 합시다.

먼저 기댓값의 정의상 $E[X] = \sum_x xP(X = x)$

예를 들어 공정한 6면 주사위의 기댓값 = $1(1/6)$
 $+ 2(1/6) + 3(1/6) + 4(1/6) + 5(1/6) + 6(1/6)$
 $= 3.5$

$$E[X + Y] = \sum_x \sum_y [(x + y) P(X = x, Y = y)]$$

$$E[X + Y] = \sum_x \sum_y [x P(X = x, Y = y)] + \sum_x \sum_y [y P(X = x, Y = y)]$$

$$E[X + Y] = \sum_x x \sum_y P(X = x, Y = y) + \sum_x y \sum_y P(X = x, Y = y)$$

$$\sum_y P(X = x, Y = y) = P(X = x)$$

$$\sum_x P(X = x, Y = y) = P(Y = y)$$

$$E[X + Y] = \sum_x x P(X = x) + \sum_x y P(Y = y)$$

$$\therefore E[X + Y] = E[X] + E[Y]$$

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

$nCr \bmod p$ for a prime p ($n, r \leq p$) in $O(n)$ Precomputation and $O(1)$ Query

1. 제목만 보면 “그게 뭔데 씹덕아“ 라고 하실수 있지만
잊을 때 되면 한번 나오는 유형의 문제입니다.
2. 정수론 지식과 조합론 지식을 적절히 섞어서 사용하는
만큼 오늘의 마지막 주제로 굉장히 적절합니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

$nCr \bmod p$ for a prime p ($n, r \leq p$) in $O(n)$ Precomputation and $O(1)$ Query

제목을 차례대로 끊어서 이해해봅시다.

1. nCr 즉, 조합을 계산합니다
2. $\bmod p$ 조합을 p 로 나눈 나머지를 계산합니다
3. For a prime p 즉, 소수 p 에 대하여
4. ($n, r \leq p$) n 과 r 이 모두 p 보다 작을때
5. In $O(n)$ Precomputation 즉, $O(n)$ 시간의 전처리로 답을 구해놓고
6. And $O(1)$ Query 전처리 해둔 답을 n 과 r 이 쿼리로 주어질때 $O(1)$ 시간에 출력합니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

우선 답을 구하는 원리부터 짧게 설명을 해보겠습니다.

$$nC_r = \frac{n!}{(n-r)!r!}$$

위는 조합을 구하는 식입니다. 그리고 우리는 여기에 모듈러 연산을 적용 시킬 겁니다. 따라서 이러한 분수 꼴의 형태를 사용하지 않고 모듈러 곱셈 역원을 사용해

$$nC_r = n! \cdot ((n-r)!)^{-1} \cdot (r!)^{-1}$$

로 나타낼수 있으며 여기서 a^{-1} 은 a 의 역원을 나타냅니다.

따라서 우리는 n 까지의 팩토리얼 값의 곱셈 역원을 $O(n)$ 시간에 전처리 할수 있다면, 각 조합 쿼리를 $O(1)$ 시간에 처리 할수 있게 됩니다.

전처리 방법으로는 간단하지만 강력한 발상을 활용합니다. 우선, $n! \bmod p$ 를 $O(n)$ 시간에 구합니다. 그 후 $n!$ 의 곱셈 역원만 먼저 확장 유클리드 알고리즘으로 계산해줍니다.

$n!$ 의 곱셈 역원을 계산했다면 여기서부터는 역순으로 역원 값들을 채워나갈건데 바로 $(n-1)!$ 의 역원은 $n!$ 의 역원과 n 의 곱임을 활용합니다. 이 두 아이디어를 통해 $n!$ 까지의 팩토리얼의 역원 을 $O(n)$ 시간에 구할 수 있습니다.

#고급 정수론, 조합론

<Advanced Number Theory and Combinatorics>

5 이항 계수와 쿼리 (BOJ #13977)

<문제 설명>

- N 이 최대 4,000,000 으로 주어질때 $nCr \bmod p$ 를 쿼리로 출력하는 문제입니다
- 여기서 p 는 1,000,000,007로 n, r 보다 큰 소수라는 조건을 만족합니다
- 계산 중간 값에 음수가 나올수 있습니다. 하지만 조합은 정의상 음수는 답이 될수 없기에 어떻게 하면 음수가 아닌 해를 출력할수 있을지 생각해보면 좋습니다.

<제약 조건>

- $1 \leq M \leq 100,000$
- $1 \leq N \leq 4,000,000$
- $1 \leq K \leq N$

#연습 문제 도전

4 Σ (BOJ #13172)

페르마 소정리를 이용해 역원을 계산해봅시다.

2 역원(Inverse) 구하기 (BOJ #14565)

짧은 설명을 적어주시면 됩니다,

4 백남이의 여행 준비의 준비 (BOJ #23062)

제가 유희왕이라는 게임을 하는데... 거시서 여행다니는
친구들은 다 상대하기 어렵던데 여기도네요...

5 이항 계수와 퀴리 (BOJ #13977)

와! 퀴리!

5 캔디 분배 (BOJ #3955)

일반해를 사용한 확장 유클리드

5 별꽃의 세레나데 (Easy) (BOJ #26217)

가차غم 하시는분?

#연습 문제 도전

Four XOR (BOJ #21099)

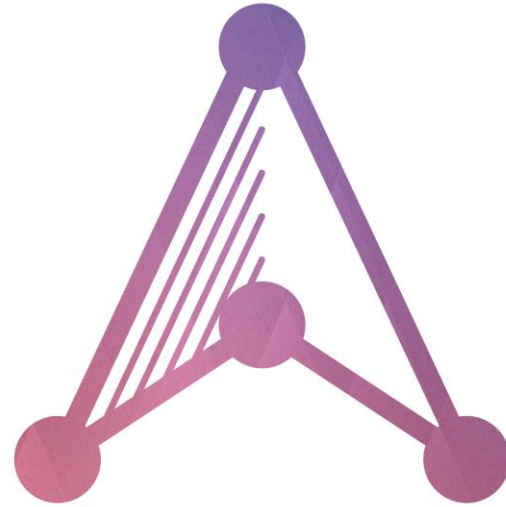
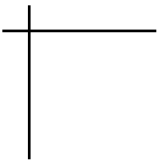
갑자기 비트 연산이?

A+B (BOJ #9267)

poj.kr/1000

BAKTERIJE (BOJ #5627)

다 풀고 심심하신 분들을 위하여!



A L O H A
The algorithm club.

