Appendix. 기초 조합론

나정휘
https://justiceHui.github.io/

목차

기본 지식

순열

순열: n개의 원소 중 서로 다른 k개의 원소를 선택해서 나열하는 방법의 수

- $n * (n-1) * \dots * (n-k+1) = n! / (n-k)!$
- 각 상자에 최대 1개의 공이 들어가도록 k개의 공을 n개의 상자에 분배하는 방법의 수
- 고등학생 때는 ${}_nP_k$ 라고 표현했음

중복 순열: n개의 원소 중 k개의 원소를 선택해서 나열하는 방법의 수

- 중복을 허용해서 k개를 선택하고 나열하는 방법
- $n * n * \dots * n = n^k$
- k개의 공을 n개의 상자에 자유롭게 분배하는 방법의 수

조합

조합: n개의 원소 중 서로 다른 k개의 원소를 선택하는 방법의 수

- {n * (n-1) * ... * (n-k+1)} / {1 * 2 * ... * k} = n! / (n-k)!k!
- 또는 C(n, k) = C(n-1, k-1) + C(n-1, k)
- 2차원 격자에서 (0, 0) → (n, m) 최단 경로의 개수는 (n+m)! / n!m! = C(n+m, n)
    - n+m번의 이동 중 세로 방향으로 이동할 n번의 위치를 정하는 경우의 수

중복 조합: n개의 원소 중 k개의 원소를 선택하는 방법의 수

- k개의 원소가 들어갈 자리를 만든 다음, n-1개의 칸막이를 원소와 원소 사이에 배치
- k+(n-1)개의 자리 중 칸막이가 들어갈 n-1개의 자리를 정하는 경우의 수
- C(n+k-1, n-1) = C(n+k-1, k)

포함 배제의 원리

교집합의 크기를 구할 수 있을 때 합집합의 크기를 구하는 방법

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

포함 배제의 원리

- 유한집합인 전체집합 U의 부분집합 $A_1, A_2, \cdots, A_n$의 합집합의 원소의 개수

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\emptyset \neq I \subset [n]} (-1)^{|I|-1} \left|\bigcap_{i \in I} A_i\right|$$

- 집합 홀수 개의 교집합 크기는 더하고, 짝수 개의 교집합 크기는 빼는 방식

질문?

페르마 소정리

합동식

합동식에서의 다양한 문제들

- 잉여역수: $ax \equiv 1\ (mod\ n)$을 만족하는 x를 찾는 문제
  - 실수 체계에서 $1/a$의 역할을 하는 값을 구하는 문제

- 이산 로그: $a^x \equiv b\ (mod\ n)$을 만족하는 x를 찾는 문제
  - 실수 체계에서 $\log_a b$의 역할을 하는 값을 구하는 문제

- 이산 제곱근: $x^2 \equiv a\ (mod\ n)$을 만족하는 x를 찾는 문제
  - 실수 체계에서 $\sqrt{x}$의 역할을 하는 값을 구하는 문제

합동식

합동식에서의 다양한 문제들

- 잉여역수
  - 만약 a의 잉여역수 a^{-1}이 존재하면, 양변을 a로 나누는 대신 a^{-1}을 곱할 수 있음
  - 페르마의 소정리를 이용하면 n이 소수이고 a가 n의 배수가 아닐 때 잉여역수를 빠르게 구할 수 있음

페르마의 소정리

페르마의 소정리

- p가 소수이고 a가 p의 배수가 아니면 $a^{p-1} \equiv 1(mod\ p)$가 성립한다.
- p가 소수이면 모든 정수 a에 대해 $a^p \equiv a(mod\ p)$가 성립한다.

- 증명
  - $a, 2a, \cdots, (p-1)a$는 $mod\ p$에서 서로 합동이 아님
    - $1 \leq i, j < p$일 때 $ia \equiv ja\ (mod\ p)$이기 위해서는 $i = j$가 되어야 함
  - 따라서 $a, 2a, \cdots, (p-1)a$를 각각 p으로 나눈 나머지들의 집합은 $\{1, 2, \cdots, p-1\}$
  - $a \times 2a \times \cdots \times (p-1)a \equiv 1 \times 2 \times \cdots \times (p-1)\ (mod\ p)$
  - $(p-1)!\, a^{p-1} \equiv (p-1)!\ (mod\ p)$
  - $(p-1)!$은 p와 서로소, 따라서 $a^{p-1} \equiv 1\ (mod\ p)$
  - a가 p의 배수가 아니면 $a^{p-1} \equiv 1\ (mod\ p)$이므로 양변에 a를 곱하면 $a^p \equiv a\ (mod\ p)$
  - a가 p의 배수면 $a^p \equiv a \equiv 0\ (mod\ p)$

페르마의 소정리

페르마의 소정리

- $a^{p-1} \equiv 1\ (mod\ p)$
- $a \times a^{p-2} \equiv 1\ (mod\ p)$
- p가 소수이고 a가 p의 배수가 아니면 a^{p-2}는 a의 잉여역수

잉여역수의 활용

$\binom{n}{r}\ mod\ p$의 빠른 계산

- 목표: 적당한 전처리를 통해 $\binom{n}{r}\ mod\ p$를 매번 $O(1)$에 계산 (단, $\binom{n}{r} = \frac{n!}{r!(n-r)!}$)
- 입력으로 주어지는 n의 최댓값을 $N < p$라고 하자.

- 전처리 $O(N^2)$
  - $\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}$를 이용해 동적 계획법으로 전처리
  - 덧셈만 이용해서 계산 가능

잉여역수의 활용

$\binom{n}{r}\ mod\ p$의 빠른 계산

- 전처리 $O(N \log p)$
  - $fac[n] \coloneqq n!\ \ mod\ p$
  - $inv[n] \coloneqq (n!)^{-1}\ mod\ p = (n!)^{p-2}\ mod\ p$

  - fac 배열은 $O(N)$ 시간에 전처리 가능
  - inv 배열은 매번 거듭제곱을 계산해서 $O(N \log p)$ 시간에 전처리 가능

  - $\binom{n}{r} \equiv \frac{n!}{r!(n-r)!} \equiv fac[n] \times inv[r] \times inv[n-r]\ (mod\ p)$
  - 따라서 전처리가 끝나면 이항 계수를 $O(1)$ 시간에 계산 가능

잉여역수의 활용

$\binom{n}{r}$ $mod\ p$의 빠른 계산

- 전처리 $O(N + \log p)$
  - inv 배열 전처리를 더 빠르게!
  - $\frac{1}{n!} = \frac{1}{(n+1)!} \times (n+1)$ 이므로 $inv[n] \equiv inv[n+1] \times (n+1)\ (mod\ p)$
  - $inv[N]$만 거듭제곱을 이용해 계산하면 $inv[0], inv[1], \cdots, inv[N-1]$은 $O(N)$시간에 전처리 가능
  - 따라서 전체 전처리 시간은 $O(N + \log p)$

잉여역수의 활용

$\binom{n}{r} \; mod \; p$의 빠른 계산

```
using ll = long long;
constexpr int SZ = 4'000'000;
constexpr ll MOD = 1'000'000'007;

ll Pow(ll a, ll b){
    ll res = 1;
    for(; b; b >>= 1, a = a * a % MOD) if(b & 1) res = res * a % MOD;
    return res;
}

ll Fac[SZ+1], Inv[SZ+1];

void Init(){
    Fac[0] = 1;
    for(int i=1; i<=SZ; i++) Fac[i] = Fac[i-1] * i % MOD;
    Inv[SZ] = Pow(Fac[SZ], MOD-2);
    for(int i=SZ-1; i>=0; i--) Inv[i] = Inv[i+1] * (i+1) % MOD;
}

ll Choose(int n, int r){
    return r <= n ? Fac[n] * Inv[r] % MOD * Inv[n-r] % MOD : 0;
}
```

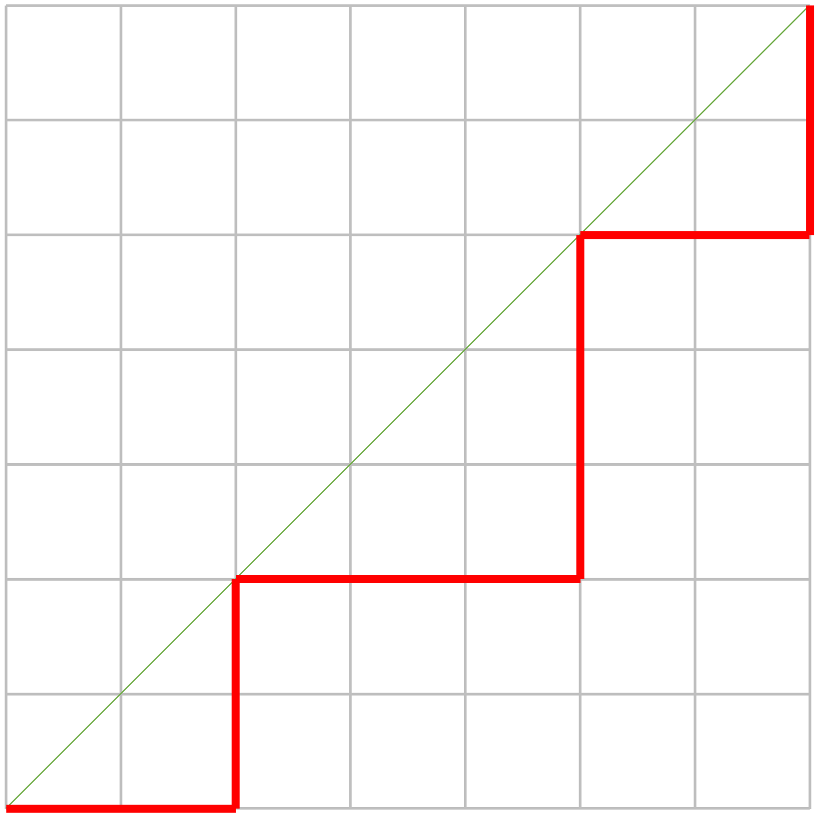
질문?

카탈란 수

카탈란 수

Dyck Path

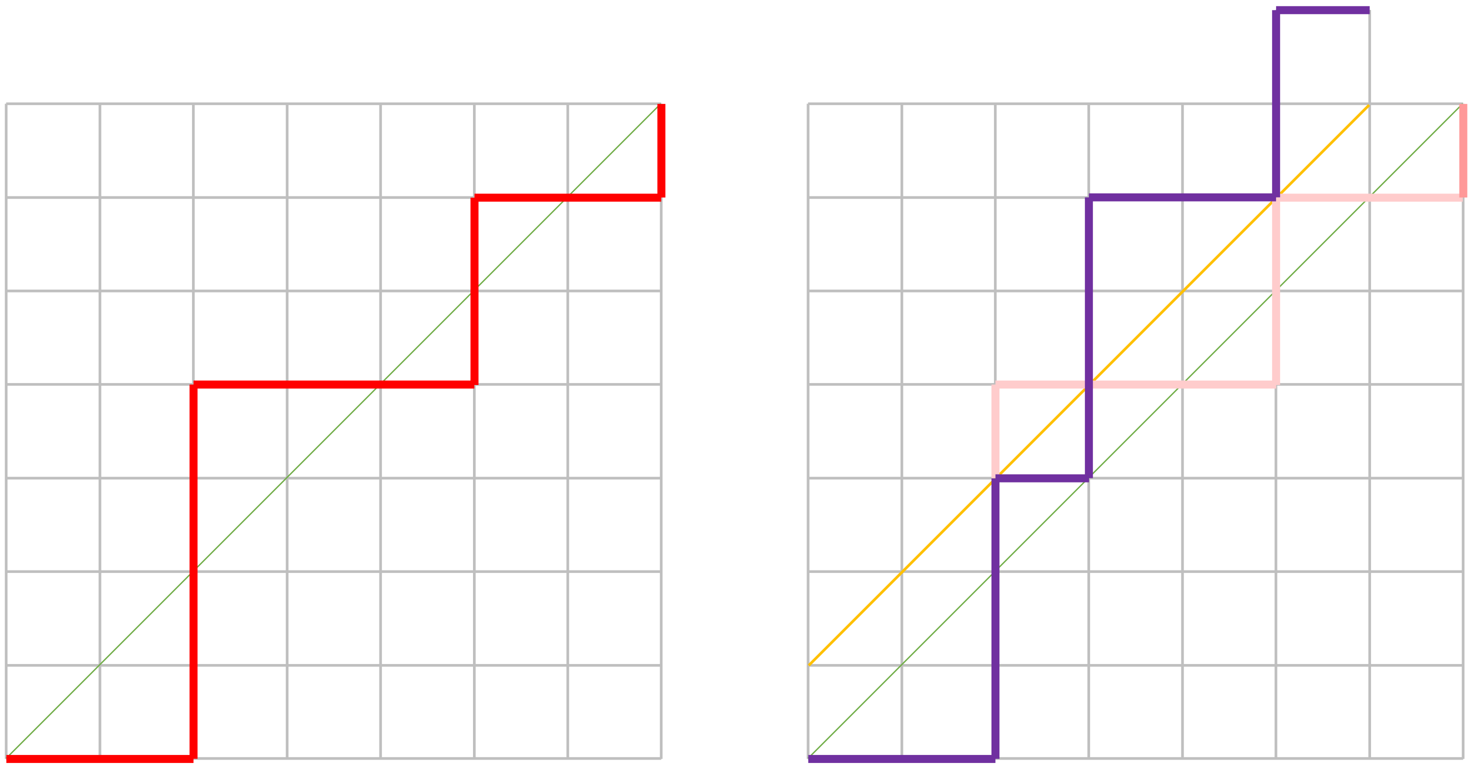
- (0, 0)에서 (n, n)으로 이동할 때, 주대각선(y = x 직선)을 넘어가지 않는 경로
- Dyck Path의 개수 = C(2n, n) – (주대각선을 한 번 이상 통과하는 경로의 개수)



카탈란 수

주대각선을 통과하는 경로의 개수

- 주대각선을 통과하는 직선은 항상 y = x+1 직선 위의 점을 방문함
- 해당 경로가 처음으로 y = x+1 과 만나는 점 이후의 모든 이동을 반전시키면?



카탈란 수

주대각선을 통과하는 경로의 개수

- 주대각선을 통과하는 직선은 항상 y = x+1 직선 위의 점을 방문함
- 해당 경로가 처음으로 y = x+1 과 만나는 점 이후의 모든 이동을 반전시키면?

카탈란 수의 일반항

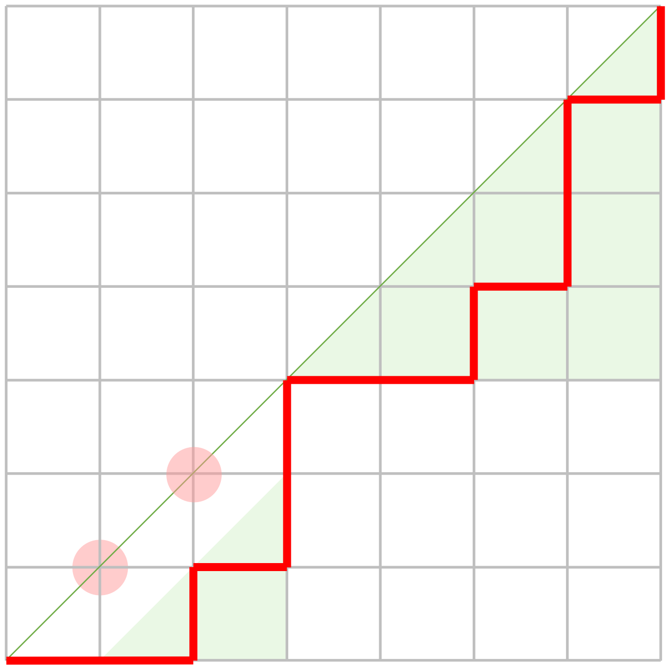
- 모든 Dyck Path는 (0, 0)에서 (n+1, n+1)로 가는 경로와 일대일 대응 가능
- 따라서 Dyck Path의 개수 = $\binom{2n}{n} - \binom{2n}{n+1}$
- 또한, $\binom{2n}{n+1} = \frac{n}{n+1}\binom{2n}{n}$이므로 $\binom{2n}{n} - \binom{2n}{n+1} = \frac{1}{n+1}\binom{2n}{n}$

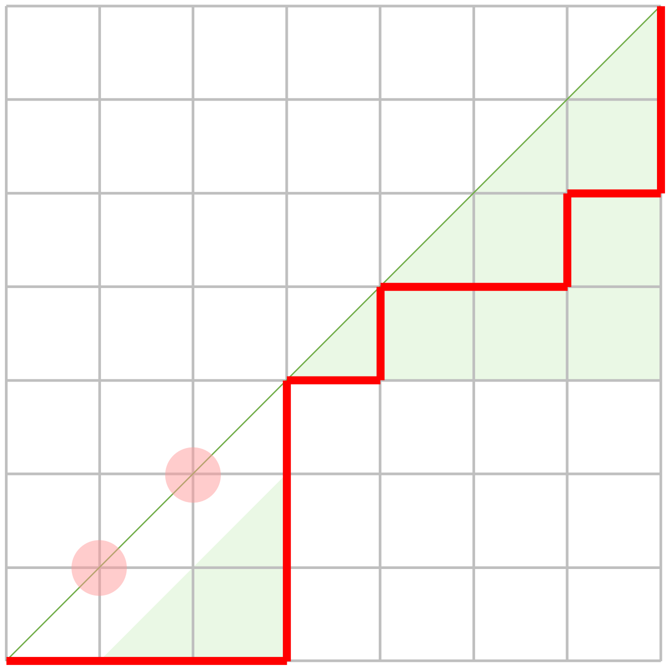
질문?

카탈란 수

카탈란 수의 점화식

- (0, 0)에서 (n, n)으로 이동하는 Dyck Path의 개수를 C_n이라고 하자.
- 원점을 제외하고 y = x 직선과 처음으로 만나는 점을 (k, k)라고 하자.
- (0, 0)에서 (k, k)까지 갈 때는 y = x 직선에 닿으면 안 됨





카탈란 수

카탈란 수의 점화식

- (0, 0)에서 (n, n)으로 이동하는 Dyck Path의 개수를 C_n이라고 하자.
- 원점을 제외하고 y = x 직선과 처음으로 만나는 점을 (k, k)라고 하자.
- (0, 0)에서 (k, k)까지 갈 때는 y = x 직선에 닿으면 안 됨

- (0, 0)에서 (k, k)로 가는 경로의 개수 = C_{k-1}
- (k, k)에서 (n, n)으로 가는 경로의 개수 = C_{n-k}

$$C_n = \sum_{k=1}^{n} C_{k-1} C_{n-k} = \sum_{k=0}^{n-1} C_k C_{n-k-1}$$

- 이 밖에도 $\sum_{n=0}^{\infty} C_n x^n = \frac{1-\sqrt{1-4x}}{2x}$ 같은 생성 함수도 있지만... 생략

질문?

카탈란 수

카탈란 수로 해결할 수 있는 문제들

- 여는 괄호 n개와 닫는 괄호 n개로 구성된 올바른 괄호 문자열의 개수
  - 여는 괄호를 ↑ 방향, 닫는 괄호를 → 방향으로 생각하면 Dyck Path와 일대일 대응

- 리프 정점이 n+1개이면서 모든 정점의 자식이 0개 또는 2개인 full binary tree의 개수
  - n+1개의 항으로 구성된 수식에 n쌍의 괄호를 씌워 이항 연산자를 적용하는 경우의 수와 동일
  - 따라서 Dyck Path와 일대일 대응

카탈란 수

카탈란 수로 해결할 수 있는 문제들

- 볼록 n+2각형에 n-1개의 대각선을 그려서 n개의 삼각형으로 분할하는 방법의 수
  - 다각형의 한 번을 고정한 뒤, 그 변을 밑변으로 하는 삼각형을 만들기 위해 대각이 될 꼭짓점을 고정하면?
  - n+2각형에서 한 변 고정했을 때 꼭짓점의 후보는 n가지
  - 그중 k(1 ≤ k ≤ n)번째 점을 선택하면 n+2각형은 다음과 같이 세 조각으로 분할됨
    - 삼각형
    - 후보가 k-1개 있는 k+1각형
    - 후보가 n-k개 있는 n-k+2각형
  - 따라서 $C_n = \sum_{k=1}^{n} C_{k-1}C_{n-k} = \sum_{k=0}^{n-1} C_k C_{n-k-1}$

질문?

집합의 분할

제2종 스털링 수

제2종 스털링 수

- n개의 원소로 구성된 집합을 k개의 공집합이 아닌 부분집합으로 분할하는 경우의 수
- ex. n = 3, k = 2면 $\{\{1,2\},\{3\}\},\{\{2,3\},\{1\}\},\{\{3,1\},\{2\}\}$으로 $S(3,2) = 3$

- n번 원소가 들어가는 집합 기준으로 생각하면 점화식을 얻을 수 있음
  - n-1개의 원소를 k-1개의 부분집합으로 분할한 뒤, 새로운 집합을 만들어서 n번 원소 삽입
  - n-1개의 원소를 k개의 부분집합으로 분할한 뒤, n번 원소를 k개의 집합 중 한 곳에 삽입
- $S(n,k) = S(n-1,k-1) + k \cdot S(n-1,k)$

벨 수

벨 수

- n개의 원소로 구성된 집합을 분할하는 경우의 수, $B_n = \sum_{k=0}^{n} S(n,k)$
  - $B_0 = S(0,0) = 1$로 맞추기 위해서 $k = 0$부터 시작
- n번 원소가 속한 부분집합의 크기를 k로 만드는 방법은?
  - n-1개의 원소 중 k-1개를 선택해서 n번 원소와 함께 하나의 집합으로 묶고
  - 남은 n-k를 적절히 분할
- $B_n = \sum_{k=1}^{n} \binom{n-1}{k-1} B_{n-k} = \sum_{k=0}^{n-1} \binom{n-1}{k} B_{n-k-1} = \sum_{k=0}^{n-1} \binom{n-1}{n-k-1} B_{n-k-1} = \sum_{k=0}^{n-1} \binom{n-1}{k} B_k$
- 두 방법 모두 $O(n^2)$ 시간에 $B_0, B_1, \cdots, B_n$ 계산 가능
- $\sum_{n=0}^{\infty} \frac{B_n}{n!} x^n = e^{e^x - 1}$ 같은 생성 함수도 있지만... 생략

제1종 스털링 수

제1종 스털링 수

- n개의 원소를 k개의 방향 있는 사이클로 분할하는 경우의 수

- n번 원소가 들어가는 사이클 기준으로 생각하면 점화식을 얻을 수 있음
  - n-1개의 원소로 k-1개의 사이클을 만든 뒤, n번 원소 하나만 있는 사이클 생성
  - n-1개의 원소로 k개의 사이클을 만든 뒤, n-1개의 간선 중 하나를 골라서 n번 원소 추가
- $S_1(n, k) = S_1(n-1, k-1) + (n-1) \cdot S_1(n-1, k)$

포함 배제의 원리

제2종 스털링 수의 일반항

- n개의 원소로 구성된 집합을 k개의 공집합이 아닌 부분집합으로 분할하는 경우의 수

- n명의 사람을 빈 방을 허용해서 k개의 구분되는 방에 배정하는 것에서 시작
  - 제2종 스털링 수는 공집합을 허용하지 않고, 집합을 구분하지 않음에 주의
  - 여기에서 빈 방이 1개 이상 생기는 모든 배정의 수를 뺀 다음 k!으로 나누면 됨

- n명의 사람을 k개의 방에 집어넣는 방법의 수 k^n
- 빈 방을 r개 이상 만드는 방법의 수 = (빈 방 r개 고정) * (n명을 k-r개의 방에 배정) = $\binom{k}{r}(k-r)^n$

- $k!\,S(n,k) = k^n + \sum_{r=1}^{k}(-1)^r\binom{k}{r}(k-r)^n = \sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n$
- $\binom{k}{r} = \binom{k}{k-r}$이므로 $k!\,S(n,k) = \sum_{r=0}^{k}(-1)^r\binom{k}{k-r}(k-r)^n = \sum_{r=0}^{k}(-1)^{k-r}\binom{k}{r}r^n$

- 따라서 $S(n,k) = \frac{1}{k!}\sum_{r=0}^{k}(-1)^{k-r}\binom{k}{r}r^n = \frac{(-1)^k}{k!}\sum_{r=0}^{k}(-1)^r\binom{k}{r}r^n$

질문?

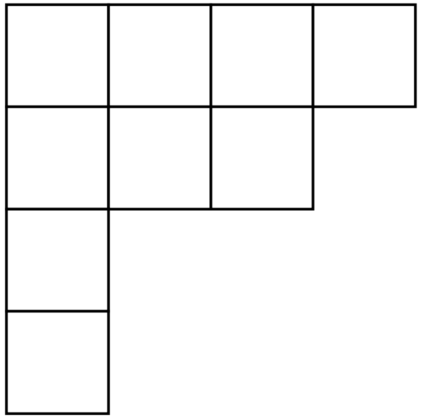
자연수의 분할

분할 수

영 다이어그램과 자연수 분할

- $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_k)$가 아래 조건을 만족할 때 λ를 N의 자연수 분할이라고 부름
  - λ_i는 양의 정수
  - $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0$
  - $\sum_{i=1}^{k} \lambda_i = N$

- 영 다이어그램: k개의 행으로 구성되어 있고, i번째 행에 λ_i개의 칸이 있는 다이어그램
- ex. $N = 9, \lambda = (4,3,1,1)$



분할 수

분할 수

- 자연수 n을 순서를 고려하지 않고 k개의 자연수의 합으로 나타내는 경우의 수
- 행이 k개인 n칸짜리 영 다이어그램

- 영 다이어그램의 마지막 행의 길이를 기준으로 생각하면
  - 마지막 행의 길이가 1인 경우: 행이 k-1개인 n-1칸짜리 영 다이어그램을 만든 뒤, 밑에 길이가 1인 행 추가
  - 2 이상인 경우: 행이 k개인 n-k칸짜리 영 다이어그램을 만든 뒤, 각 행의 길이를 1씩 증가
- 따라서 $P(n,k) = P(n-1,k-1) + P(n-k,k)$

- $P(n) = \sum_{k=1}^{n} P(n,k)$는 $O(N\sqrt{N})$에 계산할 수 있지만 어려움
  - https://infossm.github.io/blog/2021/05/18/integer-partition/

질문?

교란 순열

교란 순열

교란 순열의 점화식

- 교란 순열(완전 순열): 모든 $1 \le i \le n$에 대해 $\pi(i) \ne i$를 만족하는 길이가 n인 순열

- $p(n)$의 값을 기준으로 생각하면 D_n의 점화식을 얻을 수 있음
  - n번째 자리에 $n-1$이 들어간 상황, 즉 $p(n) = n-1$인 상황을 생각해 보자
  - n번 원소는 $p(1), p(2), \cdots, p(n-1)$ 중 원하는 자리에 들어갈 수 있음

  - 만약 n번 원소가 $n-1$번째 자리에 들어가면? $p(n-1) = n$
  - 남은 $1,2,\cdots,n-2$번 원소를 적절히 배치하면 됨
  - 정답에 D_{n-2} 만큼 기여

  - n번 원소가 $n-1$번째가 아닌 다른 자리에 들어간다면? $p(i) = n, 1 \le i < n-1$
  - n번 원소에게 $n-1$이라는 가면을 임시로 씌운 다음 $1,2,\cdots,n-1$을 이용해 교란 순열 생성
    - n번 원소가 $n-1$이라는 가면을 썼기 때문에 $n-1$번 자리로 가지 않음
    - 따라서 $p(1) = n, p(2) = n, \cdots, p(n-2) = n$인 상황을 모두 확인 가능
  - 정답에 D_{n-1} 만큼 기여

- 이를 $\pi(n) = 1,2,\cdots,n-1$ 인 상황에서 모두 확인하면 $D_n = (n-1)(D_{n-1} + D_{n-2})$

질문?

교란 순열

교란 순열의 일반항

- $D_n = (n-1)(D_{n-1} + D_{n-2}) = (n-1)D_{n-1} + (n-2)D_{n-2}$
- nD_{n-1}을 왼쪽으로 넘기면 $D_n - nD_{n-1} = -D_{n-1} + (n-1)D_{n-2}$

- $A_n = D_n - nD_{n-1}$이라고 정의하면 $A_n = -A_{n-1}$이므로 A는 공비가 -1인 등비수열
- $A_1 = D_1 - D_0 = 0 - 1 = -1$이므로 $A_n = (-1)^n$

- $A_n = D_n - nD_{n-1} = (-1)^n$, 양변을 $n!$으로 나누면 $\frac{D_n}{n!} - \frac{D_{n-1}}{(n-1)!} = \frac{(-1)^n}{n!}$
- $B_n = \frac{D_n}{n!}$으로 정의하면 $B_n - B_{n-1} = \frac{(-1)^n}{n!}$
- $B_n = B_1 + \sum_{k=2}^{n} \frac{(-1)^k}{k!}, B_1 = \frac{D_1}{1!} = 0$ 이므로 $B_n = \sum_{k=2}^{n} \frac{(-1)^k}{k!}$
- $\frac{(-1)^0}{0!} + \frac{(-1)^1}{1!} = 0$이므로 $B_n = \sum_{k=0}^{n} \frac{(-1)^k}{k!}$

- 따라서 $\mathrm{n!\,B_n} = D_n = n! \sum_{k=1}^{n} \frac{(-1)^k}{k!}$

교란 순열

교란 순열의 일반항

- 포함 배제를 사용해도 같은 점화식을 얻을 수 있음
  - n개의 원소를 나열하는 $n!$가지 방법 중 $\pi(i) = i$인 i의 개수가 k개 이상인 순열의 개수 = $\frac{n!}{k!}$
  - 따라서 $D_n = \sum_{k=0}^{n}(-1)^k \times \frac{n!}{k!} = n!\sum_{k=0}^{n}\frac{(-1)^k}{k!}$

- $\frac{D_n}{n!} = \sum_{k=0}^{n}\frac{(-1)^k}{k!}$ 은 뭔가 익숙한 형태의 식인데…

교란 순열

교란 순열의 일반항

- 포함 배제를 사용해도 같은 점화식을 얻을 수 있음
  - n개의 원소를 나열하는 $n!$가지 방법 중 $\pi(i) = i$인 i의 개수가 k개 이상인 순열의 개수 = $\frac{n!}{k!}$
  - 따라서 $D_n = \sum_{k=0}^{n}(-1)^k \times \frac{n!}{k!} = n!\sum_{k=0}^{n}\frac{(-1)^k}{k!}$

- $\frac{D_n}{n!} = \sum_{k=0}^{n}\frac{(-1)^k}{k!}$ 은 뭔가 익숙한 형태의 식인데…
  - $e^x = \sum_{k=0}^{\infty}\frac{x^k}{k!}$에 $x = -1$을 대입한 형태
  - 따라서 $\lim_{n\to\infty}\frac{D_n}{n!} = \frac{1}{e}$

질문?