제 4회 kriiicon 풀이집

kriii

A

a^x를 빠르게 구하는 문제이다.

너무 많이 쓰이는 아이디어이므로 숙지하는 것이 좋다.

다음과 같이 재귀적인 방법도 있다.

$$a^x = \begin{cases} a & ,\text{if } x = 1 \\ a^{x-1}a & ,\text{if } x > 1 \text{ and } x \text{ is odd} \\ \left(a^{\frac{x}{2}}\right)^2 & ,\text{if } x > 1 \text{ and } x \text{ is even} \end{cases}$$

Σ

$$(S_1 N_1^{-1} + S_2 N_2^{-1} + \cdots + S_M N_M^{-1}) \bmod 1{,}000{,}000{,}007$$

을 계산하는 문제이다. 역원은 문제 본문과 이전 문제를 참고하여 구하면 된다.

Ω

N칸의 계단이 있고, 당신은 계단의 가장 밑에서 부터 계단을 하나씩 올라 K번째 칸에 왔다. 이제부터 P면체 주사위를 굴려 Q 이하의 숫자가 나오면 한 칸 내려가고, 아니면 한 칸 올라간다. 계단의 정상에 오르거나, 다시 가장 밑으로 가버리면 주사위 굴리기를 멈춘다. 계단의 정상에 오를 확률을 구하여라.

예외 처리

먼저 예외 처리를 해야 한다.

```
if (K == 0) puts("0");
else if (K == N) puts("1");
else if (Q == 0) puts("1");
else if (Q == P) puts("0");
else { 나머지 }
```

나머지 부분

p_i를 현재 숫자가 i일 때 놀이가 끝나고 숫자가 N이 될 확률이라고 하면, 다음과 같은 식이 성립해야 한다.

1,2는 초기값을 나타내고 3번은 점화식을 나타낸다.

1) $p_0 = 0$

2) $p_N = 1$

3) $p_i = \frac{Q}{P} p_{i-1} + \left(1 - \frac{Q}{P}\right) p_{i+1}$

점화식 풀기

N이 작고, 식이 $N+1$개이고, 미지수가 $N-1$개이므로 가우스 소거법을 이용하여 연립방정식을 풀어내는 방법으로 문제를 해결할 수도 있다.

그러나 이러면 너무 재미 없으므로 일반항을 구하는 방법을 생각해 본다.

점화식 풀기

$\frac{Q}{P} = u$로 나타내면 점화식이

$$p_i = u\, p_{i-1} + (1-u)p_{i+1}$$

가 되고, 이 점화식의 특성방정식은

$$(1-u)x^2 - x + u = 0$$

이 되어 두 해 $x = 1$과 $x = \frac{u}{1-u}$를 얻을 수 있다.

일반항

즉, $p_i = A + B\left(\frac{u}{1-u}\right)^i$꼴로 나타난다. $\frac{u}{1-u} = v$로 나타내면

$$p_0 = A + B = 0$$

$$p_N = A + Bv^N = 1$$

이므로 $A = \frac{1}{1-v^N}$, $B = \frac{-1}{1-v^N}$이고, $p_i = \frac{1-v^i}{1-v^N}$임을 알 수 있게 된다. 이제 이것의 mod 1,000,000,007을 계산하면 된다.

일반항(2)

사실 이것으로 끝이 아니다. $u = 1$이 되어($P = 2Q$) 특성방정식에 중근이 나오는 경우가 있기 때문이다. 그렇게 되면 $p_i = A + Bi$꼴로 나타난다.

$$p_0 = A = 0$$

$$p_N = A + BN = 1$$

즉, $p_i = \frac{i}{N}$임을 알 수 있게 되었다. 역시 계산할 수 있다.

괄호

N쌍의 괄호를 사용하여 짝이 잘 맞는 괄호 문자열을 만든다. 괄호의 종류는 K개가 있다. 이 때, 원래 문자열과 문자열을 뒤집어 만든 문자열이 같은 올바른 괄호 문자열의 개수를 구하여라.

기본적인 생각

N쌍의 괄호를 사용한 괄호 문자열을 만드는 경우의 수를 격자판에서 이동하는 것으로 생각한다.

'('는 (x, y)에서 $(x+1, y)$로 이동하는 것, ')'는 (x, y)에서 $(x, y+1)$로 이동하는 것으로 생각한다.

이제 위의 두가지 이동 방법만을 이용하여 $(0,0)$에서 (N, N)으로 이동하는 경우의 수가 괄호 문자열을 만드는 경우의 수이다.

올바른 괄호 문자열($K = 1$)

현재 (x, y)에 있다면 이전에 '('가 x개, ')'가 y개 나왔다는 것이다. 그런데 $y > x$이면 괄호 쌍이 맞지 않다는 말이므로, 올바른 괄호 문자열을 만들 때는 중간에 $y > x$인 부분을 방문하면 안된다.

올바른 괄호 문자열의 수는 카탈란 넘버(Catalan number)

$C_N = \frac{1}{N+1}\binom{2N}{N}$으로 나타난다. 링크를 참조하라.

올바른 괄호 문자열$(K > 1)$

괄호의 종류가 K개가 되면 올바른 괄호 문자열의 개수는 몇 개일까?

답은 $K^N C_N$이다. N개의 괄호쌍 각각에 대해 K종류의 경우를 생각할 수 있기 때문이다.

뒤집어서 같아지는 경우의 수($K = 1$)

이제 뒤집어도 같은 올바른 괄호 문자열의 수를 센다.

뒤집어서 같아지기 위해서는 이동한 경로가 $x + y = N$에 대칭이어야 한다. 그러므로 $x + y = N$ 까지만 이동한 다음 이후에는 이전까지 이동했던 방법을 따라 이동하면 된다.

즉, $(0, 0)$에서 $x + y = N$인 점 중 하나로 최단 경로로 이동하는 경우의 수를 모두 더해주면 된다.

뒤집어서 같아지는 경우의 수($K = 1$)

$D_{x,y}$를 중간에 $y > x$인 영역을 지나지 않으면서 $(0,0)$에서 (x, y)로 이동하는 경우의 수라고 하면 다음과 같다.

$$D_{x,y} = \begin{cases} 1, & \text{if } x = y = 0 \\ 0, & \text{if } x < 0 \text{ or } y < 0 \text{ or } y > x \\ D_{x-1,y} + D_{x,y-1}, & \text{if } x \geq 1, y \geq 0, y \leq x \end{cases}$$

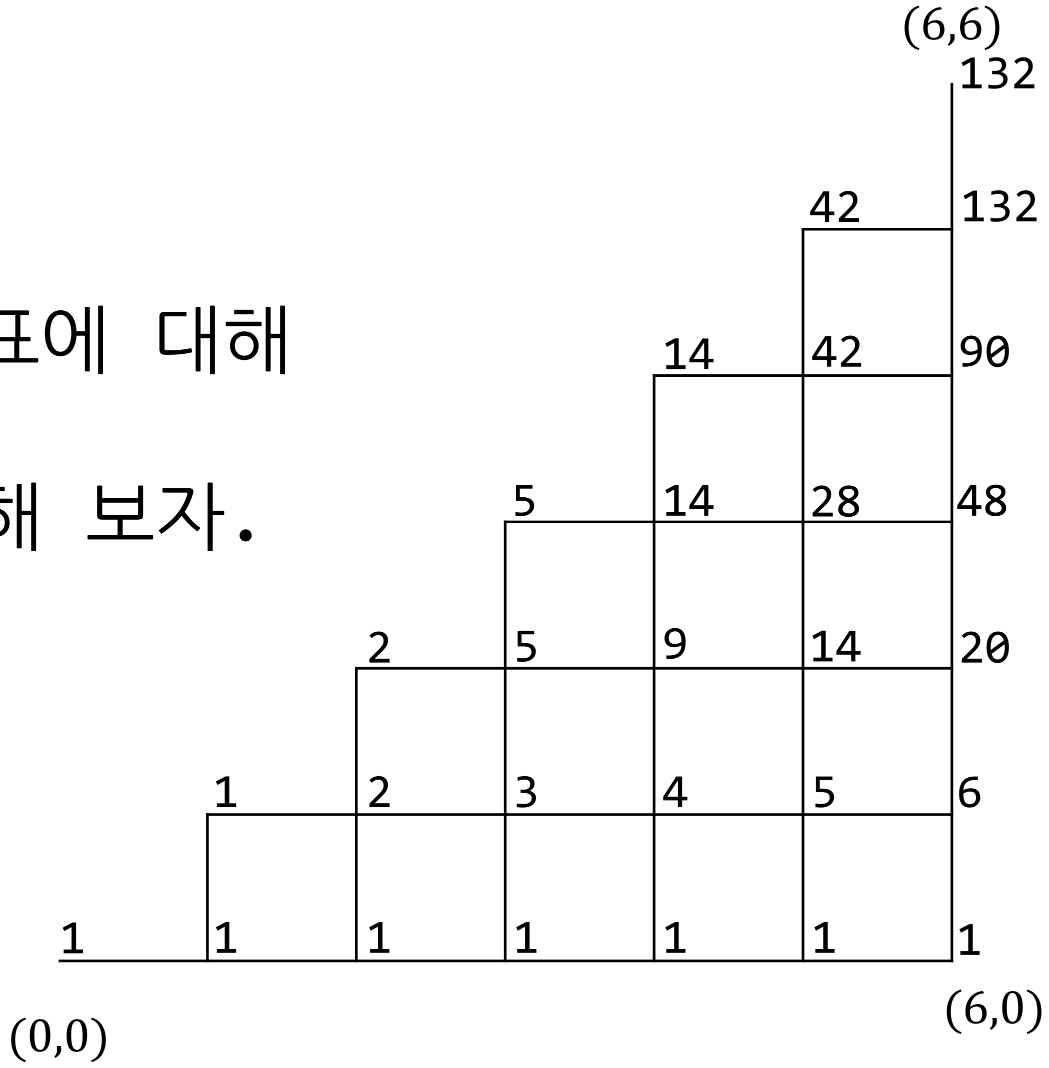
뒤집어서 같아지는 경우의 수$(K = 1)$

$D_{x,y}$를 조금 써보면 다음과 같다.

여기서 S_N을 $x + y = N$인 모든 좌표에 대해

$D_{x,y}$의 합이라고 하고 점화식을 구해 보자.

우선 $S_0 = 1$이다.



뒤집어서 같아지는 경우의 수($K = 1$)

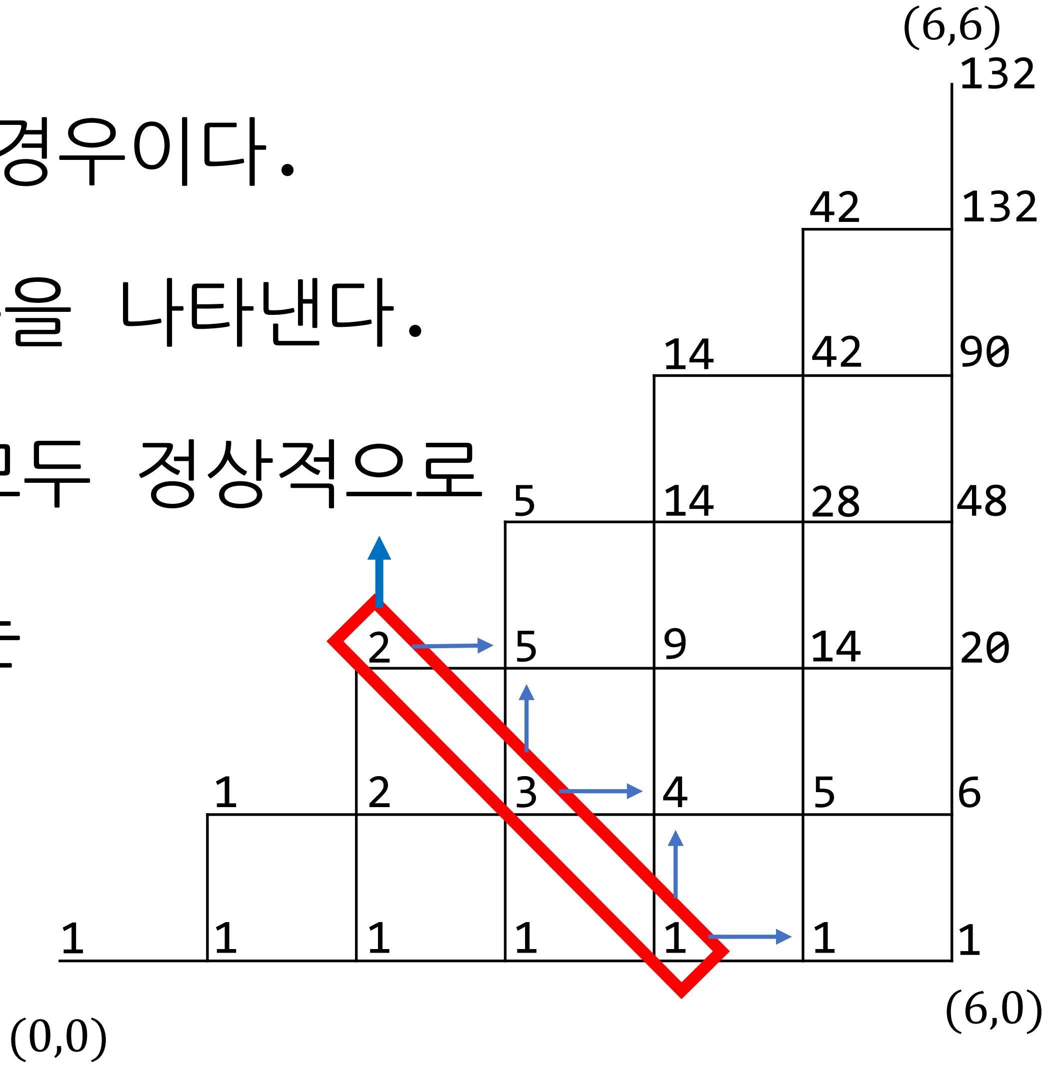
빨간색 직사각형은 $N = 4$(짝수)인 경우이다.

화살표는 각 정수가 더해지는 두 곳을 나타낸다.

(2,2)에서 위로 가는 방향 말고는 모두 정상적으로

더해진다. (2,2)로 가는 경우의 수는

이미 구했으니 빼줄 수 있다.

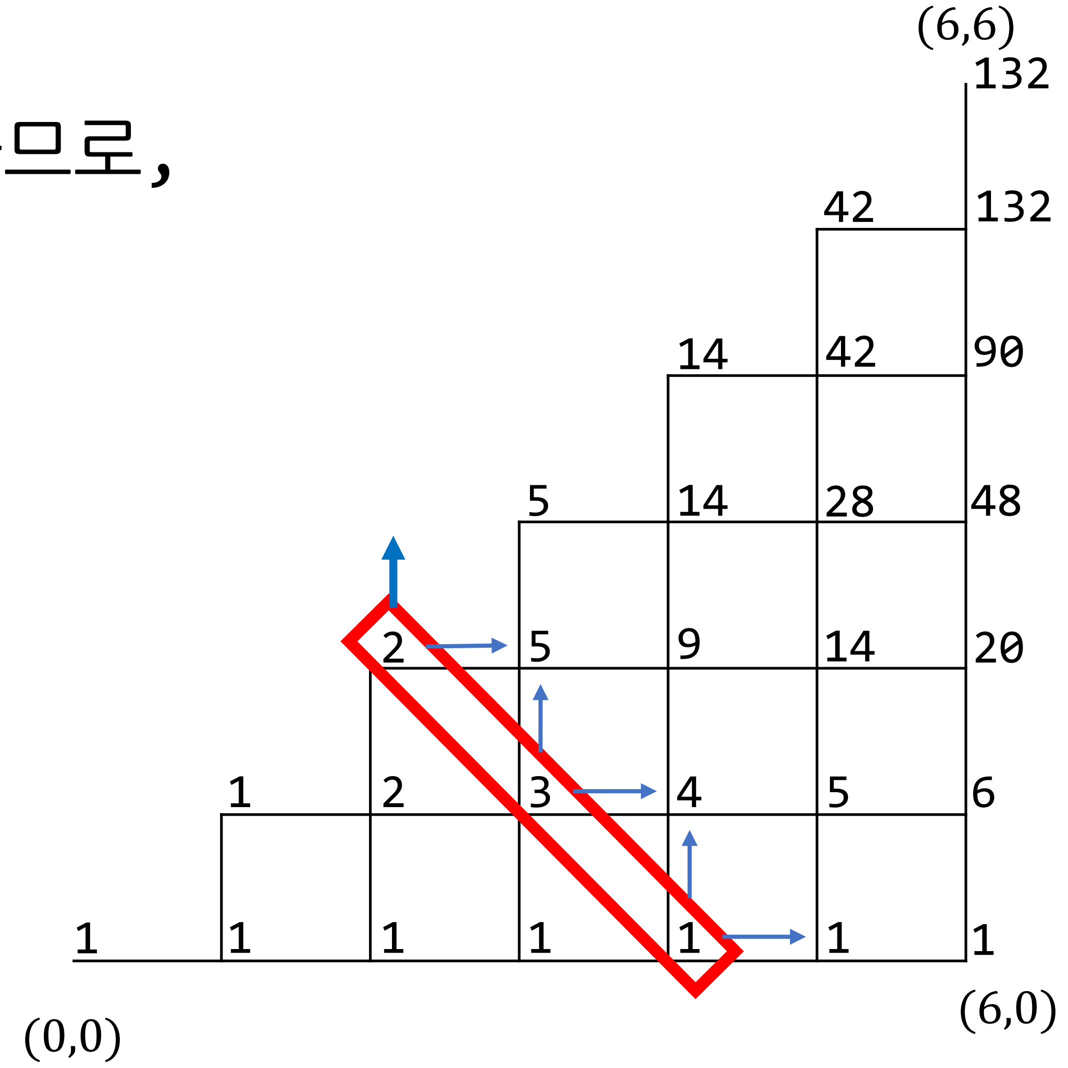


뒤집어서 같아지는 경우의 수($K = 1$)

이것은 모든 짝수 N에 대해 동일하므로,

N이 짝수이면

$S_{N+1} = 2S_N - C_{\frac{N}{2}}$가 된다.

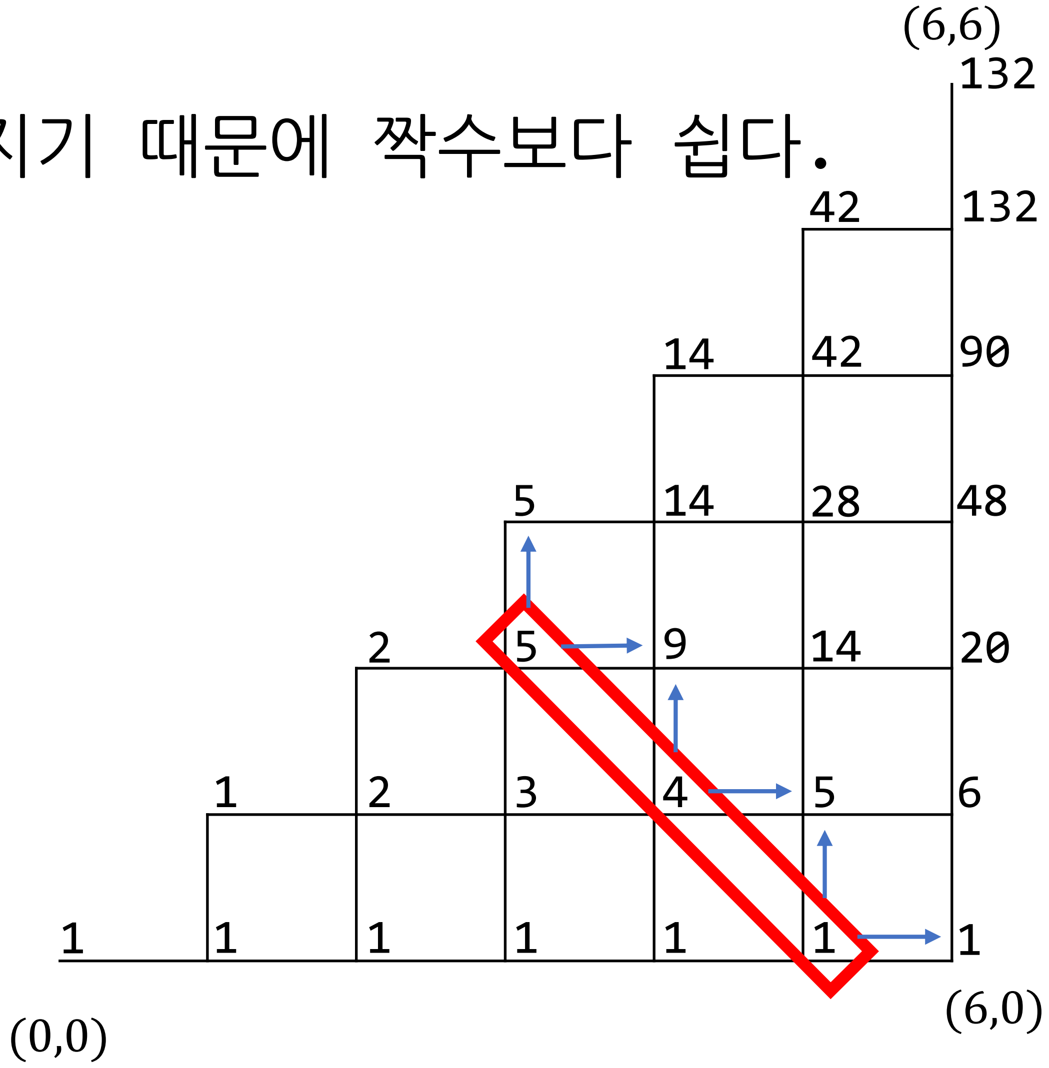


뒤집어서 같아지는 경우의 수($K = 1$)

홀수 N은 모든 수가 정상적으로 더해지기 때문에 짝수보다 쉽다.

즉, N이 홀수이면

$S_{N+1} = 2S_N$가 된다.



뒤집어서 같아지는 경우의 수($K > 1$)

$2N$개의 괄호쌍 중에서 처음 N개의 괄호 중에서 쌍이 맞는 괄호쌍이 있다면, 마지막 N개에도 이 괄호쌍에 대응하는 괄호쌍이 있을 것이고 종류가 같을 것이다. 즉 두개의 괄호 쌍이 같은 종류를 가져야 하기 때문에 자유도가 하나 줄어든다.

어떤 괄호쌍의 여는 괄호가 처음 N개의 괄호 중에 있고 닫는 괄호가 마지막 N개의 괄호 중에 있다면 별 다른 점이 없다.

뒤집어서 같아지는 경우의 수($K > 1$)

즉, $x + y = N$인 점까지 오면서 사용한 여는 괄호가 하나 늘어나면 자유도가 하나 늘어난다. 이를 이용해 $D_{x,y}$를 다음과 같이 확장할 수 있다.

$$D_{x,y} = \begin{cases} 1, & \text{if } x = y = 0 \\ 0, & \text{if } x < 0 \text{ or } y < 0 \text{ or } y > x \\ K \cdot D_{x-1,y} + D_{x,y-1}, & \text{if } x \geq 1, y \geq 0, y \leq x \end{cases}$$

뒤집어서 같아지는 경우의 수$(K > 1)$

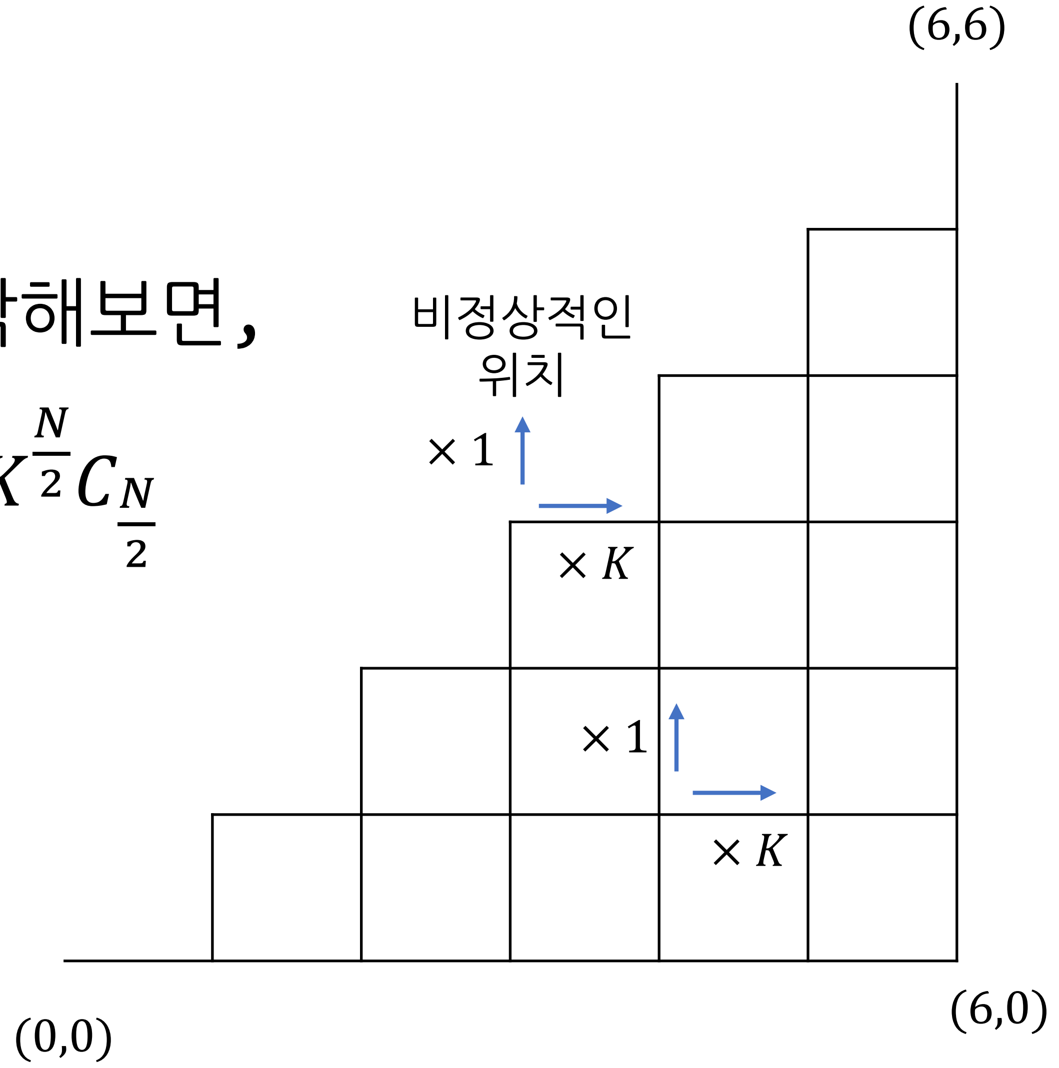
S_N도 역시 동일한 정의를 가진다.

$K = 1$일 때와 동일한 방식으로 생각해보면,

N이 짝수이면 $S_{N+1} = (K+1)S_N - K^{\frac{N}{2}} C_{\frac{N}{2}}$

N이 홀수이면 $S_{N+1} = (K+1)S_N$

이 된다.



능력

N개의 능력이 있다. 가능한 $N!$가지의 나열 중 하나를 선택하여 능력들을 순서대로 발동시킨다. 각 능력에는 발동이 성공할 확률 p_i와 성공하면 주는 피해량 d_i가 있다. 능력이 하나라도 발동하는데 성공하면 다른 능력들은 발동시키지 않고 멈춘다.

주게 되는 피해량의 기댓값을 구하여라.

순서 하나만 따질 때

사용될 능력의 순서가 미리 정해져 있다고 생각하여 이 순서를 $i_1, i_2, \ldots, i_N$이라고 하면, 이 때 피해량의 기댓값은

$$p_{i_1} d_{i_1} + \left(1 - p_{i_1}\right) p_{i_2} d_{i_2} + \cdots + \left(1 - p_{i_1}\right) \cdots \left(1 - p_{i_{N-1}}\right) p_{i_N} d_{i_N}$$

이다. 이것을 가능한 모든 순열에 대해 평균내어야 한다. 즉 $N!$로 나눠야 한다.

가능한 항의 종류

앞 페이지에 있던 식을 전개해서 모든 순열에 대해 그 합을 구했다면 존재할 수 있는 항의 종류는 정확히는 알 수 없지만 $2^N N$개를 넘을 수는 없을 것이다.

p가 곱해질 수 있는 종류가 N가지이고 뒤에 d가 하나씩 곱해져 있기 때문이다.

가능한 항의 종류

정확히 하면 다음과 같은 항들이 있을 수 있다.

$$p_1^{x_1} p_2^{x_2} \cdots p_N^{x_N} d_i$$

$$(x_1, x_2, \cdots x_N \in \{0,1\}, i \in \{1,2,\cdots,N\})$$

결론적으로는 이 항의 계수는 $\frac{(-1)^{x_1+x_2+\cdots+x_N-1}}{x_1+x_2+\cdots+x_N}$이 된다.

이걸 어떻게 알까요

사실 필자는 실제로 N이 작을 때 모두 계산해보고 일반화 하였다. 그러니 이제는 증명을 해야 할 시간이다.

능력이 N개 있고, 아직 $N!$을 나누지 않았을 때 p가 k개 곱해진 항의 계수를 $C_{N,k}$라고 하자.

$N = 1$이면 기댓값은 $p_i d_i$이므로, $C_{1,1} = 1$이다.

계수 구하기

능력이 $N-1$개에서 한 개가 추가되어 N개가 될 때 어떤 식으로 생각해야 할까?

$$p_{i_1}d_{i_1}+\left(1-p_{i_1}\right)\left[p_{i_2}d_{i_2}+\cdots+\left(1-p_{i_2}\right)\cdots\left(1-p_{i_{N-1}}\right)p_{i_N}d_{i_N}\right]$$

처럼 생각해 보자.

계수 구하기

$$\boxed{p_{i_1} d_{i_1}} + (1 - p_{i_1})\left[p_{i_2} d_{i_2} + \cdots + (1 - p_{i_2}) \cdots (1 - p_{i_{N-1}}) p_{i_N} d_{i_N}\right]$$

하나의 i_1에 대해 가능한 능력 배열의 개수는 $(N-1)!$ 이므로,

우선 $C_{N,1}$에 $(N-1)!$이 더해진다.

계수 구하기

$$p_{i_1}d_{i_1} + \boxed{(1-p_{i_1})[p_{i_2}d_{i_2} + \cdots + (1-p_{i_2})\cdots(1-p_{i_{N-1}})p_{i_N}d_{i_N}]}$$

1이 곱해져서 k개의 p를 가지게 되는 항은 i_1이 포함되지 않았을 것이므로, $N-k$배 늘어나게 된다.

$-p_{i_1}$이 곱해져서 k개의 p를 가지게 되는 항은 i_1이 다른 $k-1$개 중 하나여야 하기 때문에, $-(k-1)$배 늘어나게 된다.

즉, $C_{N,k} = (N-k)C_{N-1,k} - (k-1)C_{N-1,k-1}$ 이다.

계수 구하기

정리하면, $C_{N,1} = (N-1)! + (N-1)C_{N-1,1}$이고 $k \geq 2$이면

$C_{N,k} = (N-k)C_{N-1,k} - (k-1)C_{N-1,k-1}$ 가 된다.

$C_{N,k} = \frac{N!}{k}(-1)^{k-1}$를 넣어 테스트해보면 맞다는 것을 알 수 있다.

마지막에 $N!$을 나눠야 하므로, 계수가 $\frac{(-1)^{k-1}}{k}$임을 알게 되었.

종류가 같은 항의 합 구하기

$$p_1^{x_1} p_2^{x_2} \cdots p_N^{x_N} d_i$$

$$(x_1, x_2, \cdots x_N \in \{0,1\}, i \in \{1,2,\cdots,N\})$$

이제 항 중에서 $x_1 + x_2 + \cdots + x_N$값이 같은 것들의 합을 구하면 된다. 동적계획법을 이용하면 $O(N^2)$의 시간에 할 수 있다.

종류가 같은 항의 합 구하기

두 가지 배열이 필요하다.

1) $A[n][k]$

$\sum_{i=1}^{n} x_i = k$ 일 때 $p_1^{x_1} p_2^{x_2} \cdots p_n^{x_n}$들의 합

2) $B[n][k]$

$\sum_{i=1}^{n} x_i = k$, $1 \le i \le n,\ x^i = 1$일 때 $p_1^{x_1} p_2^{x_2} \cdots p_n^{x_n} d_i$들의 합

종류가 같은 항의 합 구하기

$$A[n][k] = A[n-1][k] + A[n-1][k-1]p_n$$

$$B[n][k] = B[n-1][k] + B[n-1][k-1]p_n + A[n-1][k-1]p_n d_n$$

$n-1$개의 능력만 고려한 상태에서 n번 능력을 추가하느냐 마느냐로 점화식을 세울 수 있다.

답

답은

$$\sum_{k=1}^{N} \frac{(-1)^{k-1}}{k} B[N][k]$$

가 된다.

동전

두 사람이 게임을 한다. 동전이 일렬로 놓여 있고, 각자의 차례에는 어떤 동전도 뒤집히지 않은 연속한 구간을 선택하여 하나 이상의 동전을 뒤집어야 한다. 어떤 동전도 뒤집을 수 없는 사람이 패배한다. 두 사람이 최선을 다 할 때 후공인 사람이 승리하게 되는 처음 동전 나열의 가짓수는 몇 가지인가?

일렬로 놓지 말고

동전을 일렬로 놓지 말고, 몇 개의 주머니에 나누어 놓았다고 생각하고, 거기에 맞게 게임을 바꾼다.

1) 하나의 주머니를 선택한다.

2) 하나 이상의 동전을 없앤다.

3) 남은 동전을 나누어 새로운 주머니들에 넣을 수 있다.

님 게임

3번 규칙을 없애면 유명한 게임인 님 게임이 된다.

1) 하나의 주머니를 선택한다.

2) 하나 이상의 동전을 없앤다.

주머니에 들어있는 동전의 개수를 모두 xor하여 0이면 그가 이길 수 있고 아니라면 이길 수 없다.

Grundy number

주머니에 들어있는 동전의 개수는 님 게임의 Grundy number이다. 그런데 신기하게도 님 게임에서 3번 규칙이 추가된다고 Grundy number가 바뀌는 일이 일어나지 않는다!

Grundy number

이는 주머니에 들어있는 동전의 개수를 N이라고 하고, 나누어 c 개의 새로운 주머니에 넣은 동전의 개수를 $x_1, \dots, x_c$라고 하면

$$x_1 \oplus \cdots \oplus x_c \leq x_1 + \cdots + x_c < N$$

이어서 어차피 mex에 변화가 없기 때문이다.

동적계획법

실제로 주머니를 구분하는 기준은 이미 뒤집힌 동전이기 때문에, C개의 동전이 들어있는 주머니는 앞면인 동전 C에 뒷면인 동전 하나가 붙어있는 형태로 나타낼 수 있다.

$D_{i,x}$를 i개의 동전을 사용했을 때, Grundy number가 x인 경우의 수라고 하여 새롭게 C개의 동전이 들어있는 주머니를 만들면 $D_{i+C+1,x\oplus C}$에 $D_{i,x}$를 더해주면 된다.

동적계획법

이 방법에서는 무조건 뒤집힌 동전을 가장 마지막에 놓기 때문에 실제로는 없는 동전 하나가 뒤집힌 상태로 있다고 생각하면 $D_{N+1,0}$가 답이 된다.

로봇

로봇이 N번 움직이려고 한다. 처음에는 x축 양의 방향을 바라보고 있다. 움직일 때는 먼저 확률적으로 왼쪽으로 90도 돌거나, 가만히 있거나 오른쪽으로 90도 돈다. 그 다음 1의 거리를 움직인다. N번 움직인 다음의 좌표가 (x, y)일 때, $x^2 + y^2$의 기댓값을 구하여라.

정답은 복소수다

로봇이 회전을 한다는 것과 구하는 것이 $x^2 + y^2$의 기댓값이라는 것에서 복소수를 떠올릴 수 있다.

즉, 로봇은 1을 $\frac{L}{L+M+R}$의 확률로, 0을 $\frac{M}{L+M+R}$의 확률로, -1을 $\frac{R}{L+M+R}$의 확률로 선택하여 이 수를 d라고 한 다음 이전에 바라보고 있는 방향 z가 있을 때, zi^d의 방향으로 이동하는 것이다.

복소수로 표현

n번째 이동에서 정해진 수를 d_n이라고 하면, 로봇이 최종적으로 이동하게 되는 칸은

$$z = x + yi = i^{d_1} + i^{d_1+d_2} + \cdots + i^{d_1+d_2+\cdots+d_N}$$

가 되고, 구하는 것은 $x^2 + y^2$이므로, $|z|^2$이다. 이것은 z의 켤레 복소수 $\bar{z}$를 이용해서 $z\bar{z}$라고 적을 수 있다.

복소수로 표현

$$\bar{z} = x - yi = i^{-d_1} + i^{-d_1-d_2} + \cdots + i^{-d_1-d_2-\cdots-d_N}$$

이므로 $z\bar{z}$는

$$z\bar{z} = \sum_{p=1}^{N}\sum_{q=1}^{N} m(p,q)\,, \qquad m(p,q) = \begin{cases} 1, & if\ p = q \\ i^{d_{q+1}+\cdots+d_p}, & if\ p > q \\ i^{-d_{p+1}-\cdots-d_q}, & if\ p < q \end{cases}$$

으로 표현가능하다.

기댓값

결국 $E(z\bar{z})$를 구하는 것이 목적이므로,

$$E(z\bar{z}) = E\left(\sum_{p=1}^{N}\sum_{q=1}^{N} m(p,q)\right) = \sum_{p=1}^{N}\sum_{q=1}^{N} E\big(m(p,q)\big)$$

를 구하면 된다.

기댓값

$$E\left(i^{d_p}\right) = u = \frac{M + (L - R)i}{L + M + R}, \qquad E\left(i^{-d_p}\right) = v = \frac{M + (R - L)i}{L + M + R}$$

이고, 각 i^{d_p}는 서로 독립적인 확률변수이므로,

$$E\left(m(p,q)\right) = \begin{cases} 1, & if\ p = q \\ u^{p-q}, & if\ p > q \\ v^{q-p}, & if\ p < q \end{cases}$$

이제 $p - q$가 같으면 기댓값이 같다는 것을 알았으니 같은 항을 묶어서 쓰면

기댓값

$$E(z\bar{z}) =$$
$$u^{N-1} + 2u^{N-2} + \cdots + (N-2)u^2 + (N-1)u^1$$
$$+N$$
$$+(N-1)v^1 + (N-2)v^2 + \cdots + 2v^{N-2} + v^{N-1}$$

으로 총 기댓값이 멱급수로 나타난다.

멱급수의 합

멱급수의 합을 식으로 나타내는 것이 그렇게 어려운 일은 아니지만 $u = 1$이거나 $v = 1$이면 예외처리를 해야하므로, 그런 것이 필요 없도록 합을 $O(\lg N)$에 구해주는 함수 $f(N)$을 생각해본다.

기본적으로 $\frac{N}{2}$일때의 합을 이용하는 아이디어를 사용한다.

$$f(a, N) = \left(A_{a,N}, B_{a,N}, C_{a,N}\right)$$
$$= \left(a^N, a^0 + a^1 + \cdots + a^{N-1}, 0 \cdot a^0 + 1 \cdot a^1 + \cdots + (N-1) \cdot a^{N-1}\right)$$

멱급수의 합

$$f(a, 0) = (1,0,0)$$

$$f(a, N+1) = (A_{a,N+1}, B_{a,N+1}, C_{a,N+1})$$
$$= \left(A_{a,N} \cdot a, B_{a,N} + A_{a,N}, C_{a,N} + N \cdot A_{a,N}\right)$$

$$f(a, 2N) = (A_{a,2N}, B_{a,2N}, C_{a,2N})$$
$$= \left(A_{a,N}^2, B_{a,N}(1 + A_{a,N}), C_{a,N}\left(A_{a,N} + 1\right) + N \cdot B_{a,N} \cdot A_{a,N}\right)$$

멱급수의 합

이제

$$E(z\bar{z}) = \left(B_{u,N} + B_{v,N}\right) \cdot N - C_{u,N} - C_{v,N} - N$$

이 된다.

목공

나무 상자를 만드는데 N개의 나무 판자가 필요하다. 나무 상자를 만들면 필요가 없으므로 분해를 한다. 분해를 하면 확률적으로 N개보다는 적은 수의 나무 판자를 얻을 수 있다. 작업은 나무 상자가 하나도 없고, 더 이상 나무 상자를 만들 수 없을 때까지 계속한다. 만들게 되는 나무 상자 개수의 기댓값을 구하여라.

이 문제의 답안은

필자가 생각한 것이 아니므로 좀 더 친절하고 자세한 풀이를 원한다면 아래의 링크를 참조하는 것을 추천한다.

http://discuss.codechef.com/problems/RNG

점화식

나무 판자가 M개일 때 만들게 되는 나무 상자 개수의 기댓값을 B_M이라고 하면,

$$B_M = \frac{q_0 B_{M-N} + q_1 B_{M+1-N} + \cdots + q_{N-1} B_{M-1}}{q_0 + q_1 + \cdots + q_{N-1}} + 1$$

이다. $B_M - B_{M-1}$을 구하면 뒤쪽의 +1이 사라진다.

점화식

이제 점화식이 선형 동차 점화식이 된다.

문제는 B_M을 구하기에는 M과 N이 크다는 사실이다.

아마도 $O(N^2 \lg M)$인 방법을 아는 사람은 많을 것이지만, $O(N \lg^2 N \lg M)$인 방법을 아는 사람은 적을 것으로 예상되어 이를 소개하고자 한다.

점화식

식을 다음과 같이 적어보자.

$$B_i = c_1 B_{i-1} + c_2 B_{i-2} + \cdots + c_k B_{i-k}$$

이 점화식의 특성방정식을 $m(x) = 0$이라하면

$$m(x) = x^k - c_1 x^{k-1} - c_2 x^{k-2} - \cdots - c_k x^0$$

이다.

점화식

이때 어떤 다항식 $f(x)$에 대해서 특성방정식이 $m(x)f(x) = 0$이 되는 점화식은 원래 점화식으로 구한 항들에 대해서도 점화식을 만족한다는 것을 증명할 수 있다.

$f(x) = a_l x^l + a_{l-1} x^{l-1} + \cdots + \cdots + a_0 x^0$이라 하면

점화식

$$m(x)f(x) =$$

$$a_l\left(x^{l+k} - c_1x^{l+k-1} - c_2x^{l+k-2} - \cdots - c_kx^{l+0}\right) + \cdots$$

$$+a_0\left(x^{0+k} - c_1x^{0+k-1} - c_2x^{0+k-2} - \cdots - c_kx^{0+0}\right) = 0$$

이므로, 모든 x^p를 b_p로 치환하면 점화식도 성립하게 되는 것을 알 수 있다.

계산해야 하는 항 개수 유지시키기

여기서 $f(x)$를 잘 선택하면, 점화식에서 계수가 0이 아닌 항을 k개 이하로 계속해서 유지시킬 수 있다.

이 $f(x)$는 $m(-x)$혹은 $-m(-x)$이다. 최고차항의 계수가 1이 되도록 둘 중 하나를 선택하면 된다. $m(x)m(-x)$가 우함수이기 때문에, 짝수차항만이 남기 때문이다.

범위 줄이기

어떤 다항식 $g(x)$를 $m(x)$로 나눈 몫을 $q(x)$, 나머지를 $r(x)$라고 하자. 즉,

$$g(x) = q(x)m(x) + r(x)$$

이다. 이제 이 방정식의 x^p를 b_p로 치환했다고 해보자. 양 변의 계수는 같으므로 등식은 유지된다. 그리고 이전 페이지에서 $q(x)m(x)$에서 x^p를 b_p로 치환하면 0이 됨을 알 수 있었다.

범위 줄이기

$S(M, m(x))$를 $b_M = s_0 b_0 + s_1 b_1 + \cdots + s_{k-1} b_{k-1}$이 되는 s열을 반환해주는 함수라고 하자. $m'(x)$를 $m(x)m(-x)$ 혹은 $-m(x)m(-x)$를 적절히 선택하고 차수를 두배로 압축한 다항식이라고 하자. $S\left(\left\lfloor \frac{M}{2} \right\rfloor, m'(x)\right)$ 는 M의 홀짝에 따라 $b_M = s'_1 b_1 + s'_3 b_3 + \cdots + s'_{2k-1} b_{2k-1}$혹은 $b_M = s'_0 b_0 + s'_2 b_2 + \cdots + s'_{2k-2} b_{2k-2}$가 되는 s'열을 인덱스가 두배로 압축된 형태로 반환해 줄 것이다.

범위 줄이기

예를 들어 $b_M = s'_0 b_0 + s'_2 b_2 + \cdots + s'_{2k-2} b_{2k-2}$가 왔다고 해보자. 그러면 방정식 $s'_{2k-2} x^{2k-2} + \cdots + s'_2 x^2 + s'_0 x$를 $m(x)$로 나눈 나머지를 구하면 $s_{k-1} x^{k-1} + \cdots + s_1 x + s_0$이 될 것인데, 이 s열은

$$b_M = s_0 b_0 + s_1 b_1 + \cdots + s_{k-1} b_{k-1}$$

를 만족하게 된다.

다항식 나눗셈

이제 길이가 긴 다항식 두개를 나눴을 때의 몫(이 문제에서는 필요 없지만)과 나머지를 구해야 한다.

$$g(x) = q(x)m(x) + r(x)$$

g의 차수를 i, m의 차수를 j라고 하고 식을 다음과 같이 변형시킨다.

$$x^i g(1/x) = \left(x^{i-j} q(1/x)\right) \cdot \left(x^j m(1/x)\right) + x^{i-j+1} \left(x^{j-1} r(1/x)\right)$$

다항식 나눗셈

$x^k h(1/x)$를 $h'_k(x)$로 적으면,

$$g'_i(x) = q'_{i-j}(x) m'_j(x) + x^{i-j+1} r_{j-1}(x)$$

$$\Rightarrow g'_i(x) = q'_{i-j}(x) m'_j(x) \mod x^{i-j+1}$$

$$\Rightarrow q'_{i-j}(x) = g'_i(x) \left(m'_j(x)\right)^{-1} \mod x^{i-j+1}$$

$\left(m'_j(x)\right)^{-1}$가 무엇일지 감이 잘 안 잡힐 것이다.

다항식 나눗셈

$\left(m_j'(x)\right)^{-1}$는 $h(x)m_j'(x) = 1 \mod x^{i-j+1}$를 만족하는 다항식이라고 하자. 그런데 이런 다항식이 항상 존재할까? 우리는 $m(x)$의 최고차항의 계수를 항상 1로 유지하고 있으므로, $m_j'(x)$의 상수항은 1이다. 그러므로 다음과 같은 과정을 통해 $h(x)$를 만들어 낼 수 있다.

다항식 나눗셈

$h_0 = 1$이라고 하자. $h_0 m_i' = 1 \mod x$이다.

h_l이 $h_l m_i' = 1 \mod x^{2^l}$인 다항식일 때,

$h_{l+1} = 2h_l - m_i' h_l^2 \mod x^{2^{l+1}}$으로 정의하면,

$h_{l+1} m_i' = \left(-1 + 2m_i' h_l - {m_i'}^2 h_l^2\right) + 1 \mod x^{2^{l+1}}$

$= 1 - (m_i' h_l - 1)^2 \mod x^{2^{l+1}}$이다.

다항식 나눗셈

$m_i' h_l - 1 = 0 \mod x^{2^l}$이므로, 제곱하면 $(m_i' h_l - 1)^2 = 0 \mod x^{2^{l+1}}$,

즉, $h_{l+1} m_i' = 1 \mod x^{2^{l+1}}$이다.

다항식 나눗셈을 다항식 곱셈을 $O(\lg N)$번 하는 문제로 치환하였다. 그러므로 한번의 다항식 나눗셈은 $O(N \lg^2 N)$의 시간에 가능하다. 즉, 전체 문제를 $O(N \lg^2 N \lg M)$의 시간에 해결할 수 있게 된다.

비트

N개의 비트가 있다. 한번의 조작에는 비트를 K번 토글링한다. 모든 비트가 1이 될 때까지 하게 되는 조작 횟수의 기댓값을 구하여라.

K개의 숫자를 뽑았을 때

같은 숫자가 두 번 뽑히면 같은 수를 두 번 토글링하는 일이므로 의미가 없다.

그러므로 어떤 숫자가 뽑힌 횟수가 짝수개인지 홀수개인지가 중요하다.

K개의 숫자를 뽑았을 때

K개의 숫자를 뽑았을 때 o개의 숫자를 홀수개 뽑고, 나머지는 모두 짝수개 뽑은 경우의 수를 $M_{K,o}$라고 하자.

$$M_{K,o} = \sum_{\substack{0 \le d_1, d_2, \cdots, d_N \le K \\ d_1 + \cdots + d_N = K \\ \sum_{i=1}^{N} (d_i \bmod 2) = o}} \binom{K}{d_1, d_2, \cdots, d_N}$$

Multinomial coefficient

$$\binom{K}{d_1, d_2, \cdots, d_N}$$

는 K개의 수 중에서 1번 종류를 d_1개, 2번 종류를 d_2개, …, N번 종류를 d_N개 뽑는 경우의 수로 다음과 같다.

$$\frac{K!}{d_1!\, d_2! \cdots d_N!}$$

Multinomial coefficient의 성질

$$\binom{K}{d_1, d_2, \cdots, d_N} = \binom{K-1}{d_1 - 1, d_2, \cdots, d_N} + \cdots \binom{K-1}{d_1, d_2, \cdots, d_N - 1}$$

가 성립한다. $\binom{K-1}{d_1-1, d_2, \cdots, d_N}$는 $K-1$개의 수를 뽑았고 하나를 더 뽑았을 때 그 수가 1인 경우의 수이다.

Multinomial coefficient의 성질

이를 역으로 보면 $\binom{K}{d_1, d_2, \cdots, d_N}$는 $\binom{K+1}{d_1+1, d_2, \cdots, d_N}, \cdots, \binom{K+1}{d_1, d_2, \cdots, d_N+1}$의 N가지 수에 더해진다고 볼 수도 있다.

$M_{K,o}$에 더해진 Multinoimal들도 위와 같이 생각하면 $M_{K+1,o'}$에 모두 한꺼번에 더해지게 될 것이다.

$$M_{K+1,o} = (N-o+1)M_{K,o-1} + (o+1)M_{K,o+1}$$

$M_{K,o}$는 홀수를 o개 가지고 있으므로, 홀수 o개중 하나가 $+1$개가 되면 홀수의 개수가 하나 줄어들 것이고, 짝수 $(N-o)$개중 하나가 $+1$되면 홀수의 개수가 하나 늘어날 것이다.

즉, $o \cdot M_{K,o}$는 $M_{K+1,o-1}$에 더해지고, $(N-\ o) \cdot M_{K,o}$는 $M_{K+1,o+1}$에 더해지는 것이다.

행렬로

$$\begin{pmatrix} M_{k+1,0} \\ M_{k+1,1} \\ M_{k+1,2} \\ \vdots \\ M_{k+1,N-2} \\ M_{k+1,N-1} \\ M_{k+1,N} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ N & 0 & 2 & \cdots & 0 & 0 & 0 \\ 0 & N-1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & N-1 & 0 \\ 0 & 0 & 0 & \cdots & 2 & 0 & N \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} M_{k,0} \\ M_{k,1} \\ M_{k,2} \\ \vdots \\ M_{k,N-2} \\ M_{k,N-1} \\ M_{k,N} \end{pmatrix}$$

M으로 이루어진 형태가 같은 형태이므로

행렬로

$$\begin{pmatrix} M_{k,0} \\ M_{k,1} \\ M_{k,2} \\ \vdots \\ M_{k,N-2} \\ M_{k,N-1} \\ M_{k,N} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ N & 0 & 2 & \cdots & 0 & 0 & 0 \\ 0 & N-1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & N-1 & 0 \\ 0 & 0 & 0 & \cdots & 2 & 0 & N \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix}^{\boxed{\boldsymbol{k}}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

행렬의 제곱도 수와 같은 방식으로 빠르게 구할 수 있다.

$O(N^3 \lg k)$의 시간이 걸린다.

문제풀이는 덤

d_z를 0의 개수가 z개일 때 모든 비트를 1로 만들기 위한 조작 횟수의 기댓값이라고 하자.

0의 개수가 z개일때 한번 조작을 해서 0의 개수가 y개로 변하게 되는 경우의 수를 $c_{z,y}$라고 하면, d_z는 다음과 같이 계산된다.

$$d_z = \frac{c_{z,0}d_0 + c_{z,1}d_1 + \cdots + c_{z,N}d_N}{N^K} + 1$$

물론 $d_0 = 0$이다.

$c_{z,y}$

우선 $M_{K,o}$를 이용해 N개 중에 아무것이나 o개가 홀수인 개수가 아니라, 특정한 o개가 홀수인 개수를 구해야 한다. 이를 $A_{K,o}$라고 하자.

$M_{K,o}$는 N개중 숫자를 o개 고르는 모든 경우에 대해 대칭적으로 경우를 세었을 것이므로, $A_{K,o}$는 다음과 같다.

$$A_{K,o} = \frac{M_{K,o}}{\binom{N}{o}}$$

$c_{z,y}$

z개의 0이 있으므로, $N-z$개의 1이 있다.

뽑은 수 중에 홀수인 개수가 o개라서 o개의 비트를 변화시키게 될 때, 0을 변화시키는 개수를 x개라고 하면 1을 변화시키는 개수는 $o-x$개일 것이다. 이 경우 $y=z-x+o-x$개가 될 것이고, $c_{z,y}$에 $\binom{z}{x}\binom{N-z}{o-x}A_{K,o}$를 더해주면 될 것이다.

답

이제 만들어낸 $c_{z,y}$를 이용해 d들을 미지수로 하는 식들을 나타내는 행렬을 만들어 가우스 소거법으로 각 값이 무엇인지를 구하면 된다.

그리고 d_1에서 d_N까지를 순서대로 출력하면 된다.

순열

$N!$가지의 모든 순열에 대해 $\mathrm{K}-\mathrm{minsum}$을 구하여 그 합을 출력한다.

$$\mathrm{K}-\mathrm{minsum}(A)=\sum_{i=1}^{N}\sum_{j=i+K}^{N}\min(A_i, A_{i+1}, \ldots, A_{j-1}, A_j)$$

$j-i$가 같으면

가능한 모든 순열에 대해 $\min(A_i, A_{i+1}, \ldots, A_{j-1}, A_j)$의 합을 구하면, $j-i$가 같은 경우 순열의 나머지 원소는 아무런 영향이 없으므로 그 합이 동일할 것이다. $A_i, A_{i+1}, \ldots, A_{j-1}, A_j$의 개수 $j-i+1=l$ 이라고 하자. $l \geq K+1$인 것들에 대해 합을 구하면 된다.

$\min(A_i, A_{i+1}, \dots, A_{j-1}, A_j)$의 합

$\min(A_i, A_{i+1}, \dots, A_{j-1}, A_j) = M$인 경우의 수를 구해보자.

모든 원소가 M이상인 경우에서 $M+1$이상인 경우를 제외하면 되기 때문에, $\left\{\binom{N-M+1}{l} - \binom{N-M}{l}\right\} l!$이다.

가능한 모든 $A_i, \dots, A_j$에 대해 $\min(A_i, \dots, A_j)$의 합을 구하면 $l! \sum_{n=1}^{N}\left\{\binom{N-n+1}{l} - \binom{N-n}{l}\right\} n$ 이다. 뒤쪽의 시그마를 풀어보면

$\min(A_i, A_{i+1}, \dots, A_{j-1}, A_j)$의 합

$$\binom{N}{l} - \binom{N-1}{l} + 2\binom{N-1}{l} - 2\binom{N-2}{l}$$

$$\dots + (N-1)\binom{2}{l} - (N-1)\binom{1}{l} + N\binom{1}{l} - N\binom{0}{l}$$

$$= \sum_{n=1}^{N} \binom{n}{l} = \binom{N+1}{l+1}$$

마지막은 하키스틱 정리를 이용한 것이다.

전체합

$j-i$가 같은 것의 개수는 $N-j+i=N-l+1$개 이고, A의 나머지 부분을 정하는 경우의 수가 $(N-j+i-1)!=(N-l)!$이므로, 구해야 하는 합은

$$\sum_{l=K+1}^{N}(N-l+1)!\,l!\binom{N+1}{l+1}=(N+1)!\sum_{l=K+1}^{N}\frac{N-l+1}{l+1}$$

이다.

악수

정점이 N개 있는 그래프가 있다. 처음에는 어떤 간선도 없다.

무작위한 순서로 가능한 $\frac{N(N-1)}{2}$개의 간선을 하나씩 추가할 때,

그래프가 한 컴포넌트가 되는 시점의 기댓값은 언제인가?

구해야 하는 것

기댓값을 구하기 위해 K번째 간선을 추가할 때 그래프가 한 컴포넌트가 되는 확률을 구해야 한다.

편하기 하기 위해 확률은 마지막에 구하고 먼저 $\frac{N(N-1)}{2}$개의 간선 중에서 K개의 간선을 뽑아 순서대로 추가할 때 마지막 간선을 추가하고 나서 그래프가 한 컴포넌트가 되는 경우의 수 $C_{N,K}$를 구해보자.

구해야 하는 것

$C_{N,K}$를 구하기 위해 $D_{N,K}$를 정의한다. $D_{N,K}$는 N개의 정점을 가진 그래프에서 K개의 간선을 순서대로 추가했을 때 그래프가 한 컴포넌트가 되는 경우의 수이다.

$D_{N,K} = C_{N,K} + \left(\frac{N(N-1)}{2} - K + 1\right) D_{N,K-1}$이다.

$C_{N,K}$는 정확히 K번째에 한 컴포넌트가 되는 경우의 수이고,

$\left(\frac{N(N-1)}{2} - K + 1\right) D_{N,K-1}$는 이미 한 컴포넌트가 된 그래프에 간선을 하나 더 추가하는 경우의 수이다.

$C_{N,K}$구하기

K번째에 추가하는 시점에 그래프가 한 컴포넌트가 되려면, 이 간선은 두 컴포넌트를 연결해야한다.

그러므로 이 때 연결하는 두 컴포넌트의 크기 i, $N-i$와 두 컴포넌트에 포함된 간선의 수 x, $K-x-1$을 결정한 다음 가능한 경우의 수를 곱하고 더해주면 된다.

$C_{N,K}$구하기

$$C_{N,K} = \sum_{i=1}^{N-1} \sum_{j=0}^{K-1} \binom{N-1}{i-1} \binom{K-1}{j} D_{i,j} D_{N-i,K-j} \{i(N-i)\}$$

이다. $\binom{N-1}{i-1}$은 정점을 선택하는 경우의 수이다. (x, y)를 추가하는 것과 (y, x)를 추가하는 것이 다르지 않기 때문에, 한쪽의 컴포넌트에 한 정점이 들어가는 것을 강제해서 경우의 수가 중복되어 세어지는 것을 방지한다.

$C_{N,K}$구하기

$$C_{N,K} = \sum_{i=1}^{N-1} \sum_{j=0}^{K-1} \binom{N-1}{i-1} \binom{K-1}{j} D_{i,j} D_{N-i,K-j} \{i(N-i)\}$$

$\binom{K-1}{j}$은 간선을 선택하는 경우의 수이다. 마지막 간선을 제외한 나머지 간선을 두 컴포넌트에 분배하는 것이다.

$C_{N,K}$구하기

$$C_{N,K} = \sum_{i=1}^{N-1} \sum_{j=0}^{K-1} \binom{N-1}{i-1} \binom{K-1}{j} D_{i,j} D_{N-i,K-j} \{i(N-i)\}$$

이렇게 해서는 순서 상관없이 컴포넌트가 어떤 구성인지만 결정 되었으므로, 순서를 결정하기 위해 각 컴포넌트에 대한 순서인 $D_{i,j}$와 $D_{N-i,K-j}$를 곱해준다.

$C_{N,K}$구하기

$$C_{N,K} = \sum_{i=1}^{N-1} \sum_{j=0}^{K-1} \binom{N-1}{i-1} \binom{K-1}{j} D_{i,j} D_{N-i,K-j} \{i(N-i)\}$$

그리고 마지막으로 두 컴포넌트간을 연결하는 간선의 개수 $i(N-i)$를 곱해준다.

답

$C_{N,K}$는 K개의 간선을 선택한 경우의 수이므로, 총 $\frac{N!}{(N-K)!}$의 경우 중 하나가 선택된 것이다. 그러므로 기댓값은

$$\sum_{K=0}^{\frac{N(N-1)}{2}} \frac{C_{N,K} \cdot (N-K)!}{N!} \cdot K$$

이다.

제비

빨간색, 초록색, 파란색 제비가 들어있는 통에서 제비 뽑기를 한다. 빨간색 제비를 뽑으면 제비를 버린다. 초록색 제비를 뽑으면 통에 다시 넣는다. 파란색 제비를 뽑으면 통에 다시 넣는다. 파란색 제비를 특정 횟수 뽑게 되면 제비 뽑기를 멈춘다. 멈추게 되는 제비 뽑기 횟수의 기댓값을 구하여라.

가변적인 것

제비를 뽑는 과정에서 가변적인 것은 두 가지이다.

1) 통에 들어있는 빨간 제비의 개수(r)

2) 파란 제비를 뽑을 횟수로 남은 수(k)

$D_{r,k}$를 현재 상황에서 앞으로 뽑게 될 제비 개수의 기댓값 이라고 정의하자.

점화식

$$D_{r,k} = \frac{rD_{r-1,k} + GD_{r,k} + BD_{r,k-1}}{r+G+B} + 1$$

이고 $D_{r,k}$를 이항한 후 잘 정리하면

$$D_{r,k} = \frac{rD_{r-1,k} + BD_{r,k-1}}{r+B} + 1 + \frac{G}{r+B}$$

가 되어 이 점화식으로 $D_{R,K}$의 값을 구하면 된다.

그런데 범위가??????

그런데 R, G, B, K의 범위가 너무 크다...

일반항을 알아야 하는 것 같다.

그러므로 하나씩 계산해보고 일반항을 추측해 보도록 한다.

$G = 0$인 경우

$$D_{r,k} = \frac{rD_{r-1,k} + BD_{r,k-1}}{r+B} + 1$$

당연히 $D_{r,0} = 0, D_{0,k} = k$이다.

이제 $r = 1$인 경우를 생각해 보면 점화식은 아래와 같다.

$$D_{1,k} = \frac{k + BD_{1,k-1}}{1+B} + 1$$

$G = 0, r = 1$인 경우

$$D_{1,0} = 0, \qquad D_{1,1} = \frac{1}{1+B} + 1$$

$$D_{1,2} = \frac{2 + B\left(\frac{1}{1+B} + 1\right)}{1+B} + 1 = \frac{2(1+B) + B(1+1+B)}{(1+B)^2} + 1$$

$$= \frac{B^2 + 4B + 2}{(1+B)^2} + 1 = \frac{2B+1}{(1+B)^2} + 2$$

아직 알 수가 없다...

$G = 0, r = 1$인 경우

$$D_{1,3} = \frac{3 + B\left(\frac{2B+1}{(1+B)^2} + 2\right)}{1+B} + 1 = \frac{2B^3 + 9B^2 + 9B + 3}{(1+B)^3} + 1$$

$$= \frac{3B^2 + 3B + 1}{(1+B)^3} + 3 = \frac{(1+B)^3 - B^3}{(1+B)^3} + 3 = 4 - \left(\frac{B}{1+B}\right)^3$$

이제 뭔가 알 것같은 형태가 나왔으니 한번 테스트를 해 보았다.

$D_{1,1}, D_{1,2}$ 에서도 같은 형식이 나오니 가능성이 높다고 할 수 있다.

$G = 0$이면 $D_{1,k} = 1 + k - \left(\frac{B}{1+B}\right)^k$?

위의 가정을 $D_{1,k} = \frac{k+BD_{1,k-1}}{1+B} + 1$에 대입해보면

$$\frac{k + BD_{1,k-1}}{1+B} + 1 = \frac{k + B\left(1 + k - 1 - \left(\frac{B}{1+B}\right)^{k-1}\right)}{1+B} + 1$$

$$= \frac{k + Bk - B\left(\frac{B}{1+B}\right)^{k-1}}{1+B} + 1 = -\left(\frac{B}{1+B}\right)^k 1 + k = D_{1,k}$$

맞았다..!

$G = 0$일 때 $D_{2,k}$도 알아보기

$$D_{2,k} = \frac{2D_{1,k} + BD_{2,k-1}}{2+B} + 1 = \frac{2\left(1 + k - \left(\frac{B}{1+B}\right)^k\right) + BD_{2,k-1}}{2+B} + 1$$

이므로

$$D_{2,1} = \frac{2\left(2 - \frac{B}{1+B}\right)}{2+B} + 1 = \frac{2\left(\frac{2+B}{1+B}\right)}{2+B} + 1 = \frac{2}{1+B} + 1$$

이다. 분모의 $2+B$가 사라졌음이 보인다.

$G=0$일 때 $D_{2,k}$도 알아보기

$$D_{2,2}=\frac{2\left(3-\left(\frac{B}{1+B}\right)^2\right)+B\left(\frac{2}{1+B}+1\right)}{2+B}+1$$

$$=\frac{6+B+\frac{2B}{1+B}-2\left(\frac{B}{1+B}\right)^2}{2+B}+1=\frac{4+\frac{2B}{(1+B)^2}}{2+B}+2$$

$$=\frac{\frac{4B^2+10B+4}{(1+B)^2}}{2+B}+2=\frac{\frac{2(2B+1)(B+2)}{(1+B)^2}}{2+B}+2=\frac{2(2B+1)}{(1+B)^2}+2$$

$G = 0$일 때 $D_{2,k}$도 알아보기

$$D_{2,3} = \frac{2\left(4 - \left(\frac{B}{1+B}\right)^3\right) + B\left(\frac{2(2B+1)}{(1+B)^2} + 2\right)}{2+B} + 1$$

$$= \frac{8 + 2B + \left(\frac{4B^2 + 2B}{(1+B)^2} - \frac{2B^3}{(1+B)^3}\right)}{2+B} + 1$$

$$= \frac{\frac{6B^3 + 18B^2 + 14B + 4}{(1+B)^3}}{2+B} + 3 = \frac{2\frac{(2+B)(3B^2+3B+1)}{(1+B)^3}}{2+B} + 3$$

$G = 0$일 때 $D_{2,k}$도 알아보기

$$D_{2,3} = 2\frac{3B^2 + 3B + 1}{(1+B)^3} + 3 = 2\left(1 - \left(\frac{B}{1+B}\right)^3\right) + 3$$

$$D_{2,2} = \frac{2(2B+1)}{(1+B)^2} + 2 = 2\left(1 - \left(\frac{B}{1+B}\right)^2\right) + 2$$

$$D_{2,1} = \frac{2}{1+B} + 1 = 2\left(1 - \frac{B}{1+B}\right) + 1$$

역시 $r = 1$일 때와 비슷한 규칙이 보이므로 진도를 확 늘려보자.

$$G = 0\text{이면 } D_{r,k} = k + r\left(1 - \left(\frac{B}{1+B}\right)^{k}\right)?$$

위의 가정을 $D_{r,k} = \frac{rD_{r-1,k} + BD_{r,k-1}}{r+B} + 1$에 대입해보면.

$$\frac{rD_{r-1,k} + BD_{r,k-1}}{r+B} + 1$$

$$= \frac{r\left(k + (r-1)\left(1 - \left(\frac{B}{1+B}\right)^{k}\right)\right) + B\left(k - 1 + r\left(1 - \left(\frac{B}{1+B}\right)^{k-1}\right)\right)}{r+B} + 1$$

$G = 0$이면 $D_{r,k} = k + r\left(1 - \left(\frac{B}{1+B}\right)^{k}\right)$?

$$= \frac{r\left(k + (r-1)\left(1 - \left(\frac{B}{1+B}\right)^{k}\right)\right) + B\left(k - 1 + r\left(1 - \left(\frac{B}{1+B}\right)^{k-1}\right)\right)}{r+B} + 1$$

$$= \frac{rk + r(r-1) + B(k-1) + Br - r(r-1)\left(\frac{B}{1+B}\right)^{k} - Br\left(\frac{B}{1+B}\right)^{k-1}}{r+B} + 1$$

$G = 0$이면 $D_{r,k} = k + r\left(1 - \left(\frac{B}{1+B}\right)^{k}\right)$?

$$= \frac{rk + r(r-1) + B(k-1) + Br - r(r-1)\left(\frac{B}{1+B}\right)^{k} - Br\left(\frac{B}{1+B}\right)^{k-1}}{r+B} + 1$$

$$= \frac{(r+B)k + (r+B)r - r - B - r(r-1)\left(\frac{B}{1+B}\right)^{k} - Br\left(\frac{B}{1+B}\right)^{k-1}}{r+B} + 1$$

$$= k + r - \frac{\frac{r-1}{1+B} + 1}{r+B} rB\left(\frac{B}{1+B}\right)^{k-1} = k + r - \frac{r - 1 + 1 + B}{(r+B)} r\left(\frac{B}{1+B}\right)^{k}$$

$G = 0$이면 $D_{r,k} = k + r\left(1 - \left(\frac{B}{1+B}\right)^k\right)$?

$$= k + r - \frac{r - 1 + 1 + B}{(r + B)} r \left(\frac{B}{1 + B}\right)^k = k + r - \frac{r + B}{(r + B)} r \left(\frac{B}{1 + B}\right)^k$$

$$= k + r - r\left(\frac{B}{1 + B}\right)^k = k + r\left(1 - \left(\frac{B}{1 + B}\right)^k\right)$$

이제 $G = 0$인 경우를 완전히 해결하게 되었다.

$G > 0$인 경우

$$D_{r,k} = \frac{rD_{r-1,k} + BD_{r,k-1}}{r+B} + 1 + \frac{G}{r+B}$$

뒤쪽에 1만 있는 경우를 해결하였으니 1이 없는 경우의 해를 구해서 둘을 더하면 된다. 즉

$$D_{r,k} = \frac{rD_{r-1,k} + BD_{r,k-1}}{r+B} + \frac{G}{r+B}$$

를 해결하면 된다.

$G > 0$인 경우

이미 이런 점화식을 해결하는 데에 이골이 난 여러분은 바로 답이 무엇인지 알아차렸을 것이다. 필자가 지쳤기 때문이 아니다.

$$D_{r,k} = \frac{Gk}{B} = \frac{rD_{r-1,k} + BD_{r,k-1}}{r+B} + \frac{G}{r+B}$$

$$= \frac{rk + B(k-1)}{(r+B)B}G + \frac{G}{r+B} = \frac{Gk}{B} - \frac{G}{r+B} + \frac{G}{r+B} = \frac{Gk}{B}$$

네, 그렇습니다. 여러분 최고!

최종

즉,

$$D_{r,k} = \frac{rD_{r-1,k} + BD_{r,k-1}}{r+B} + 1 + \frac{G}{r+B}$$

의 일반항은

$$D_{r,k} = k + r\left(1 - \left(\frac{B}{1+B}\right)^k\right) + \frac{Gk}{B}$$

입니다. 축하합니다.

창문

벽돌로 이루어진 직사각형의 일부를 뚫는데, 뚫는 위치를 무작위로 정한다. 뚫게 되는 벽돌 개수의 기댓값을 구하여라.

직접 합을 구한다

일단 모든 경우에 대해 비용의 합을 구한다. 9는 나중에 곱한다.

$$\sum_{i=1}^{H}\sum_{j=i}^{H}\sum_{k=1}^{W}\sum_{l=i}^{W}(j-i+1)(l-k+1)$$

$$=\sum_{i=1}^{H}\sum_{j=i}^{H}(j-i+1)\sum_{k=1}^{W}\sum_{l=i}^{W}(l-k+1)$$

$$=\left[\sum_{i=1}^{H}\sum_{j=i}^{H}(j-i+1)\right]\times\left[\sum_{k=1}^{W}\sum_{l=i}^{W}(l-k+1)\right]$$

한 축에 대해서

$$\sum_{i=1}^{H}\sum_{j=i}^{H}(j-i+1)=\sum_{i=1}^{H}\sum_{j=1}^{H-i+1}j=\sum_{i=1}^{H}\frac{(H-i+1)(H-i+2)}{2}=\sum_{i=1}^{H}\frac{i(i+1)}{2}$$

$$=\sum_{i=1}^{H}\frac{i(i+1)(i+2-i+1)}{2\times 3}=\sum_{i=1}^{H}\frac{i(i+1)(i+2)-(i-1)i(i+1)}{6}$$

한 축에 대해서

$$\sum_{i=1}^{H} \frac{i(i+1)(i+2) - (i-1)i(i+1)}{6}$$

망원급수 이므로,

$$= \frac{H(H+1)(H+2)}{6}$$

기댓값

즉, 비용의 총합은

$$9\frac{H(H+1)(H+2)}{6}\frac{W(W+1)(W+2)}{6}$$

이고, 가능한 경우의 수는

$$\frac{H(H+1)}{2}\frac{W(W+1)}{2}$$

이므로 둘을 나누면 기댓값이 된다.

기댓값

즉 답은

$$(H+2)(W+2)$$

이다.

$$\frac{9\,\frac{H(H+1)(H+2)}{6}\,\frac{W(W+1)(W+2)}{6}}{\frac{H(H+1)}{2}\,\frac{W(W+1)}{2}}$$

을 모듈러 곱셈의 역원으로 계산하려고 하면 곤란할 수 있다…

카드

N종류의 카드가 있는 게임이 있다. L장의 카드를 샀다. 각 카드가 무작위하게 선택된다고 할 때, i번 카드를 d_i개 이상씩 뽑았을 확률은 얼마인가?

결국

$$\frac{1}{N^L} \sum_{\substack{D_1 \le C_1, \cdots, D_N \le C_N \\ C_1 + \cdots + C_N = L}} \binom{L}{C_1, C_2, \cdots C_N}$$

을 구하는 문제이다.

방법

소개할 방법은 두 가지이다.

1) FFT 사용

2) Stirling number 사용

FFT 사용

식의 Multinomial에서 $L!$을 꺼낸다.

$$\frac{L!}{N^L} \sum_{\substack{D_1 \le C_1, \cdots, D_N \le C_N \\ C_1 + \cdots + C_N = L}} \frac{1}{C_1!\, C_2! \cdots C_N!}$$

$\frac{L!}{N^L}$을 제외한 나머지 부분의 생성함수를 생각해보자.

어렵게 생성함수라는 말을 썼지만 간단한 동적계획법을 한다고 생각해도 다음 페이지와 같은 결론을 이끌어낼 수 있다.

FFT 사용

$$f_i(x) = \sum_{j=D_i}^{\infty} \frac{x^j}{j!}$$

라고 하고,

$$g(x) = \prod_{i=1}^{N} f_i(x)$$

라고 하면, 우리가 구하는 수는 $g(x)$에서 x^L의 계수임을 알 수 있다. 실제로는 차수가 L보다 작은 항만 고려하면 된다.

FFT 사용

f_i들을 곱하면 FFT를 사용해도 $O(NL\lg L)$의 시간이 걸려 느리다.
그렇기 때문에, D_i종류가 11가지라는 것을 이용하도록 한다.
$D_i = k$인 것들의 개수를 c_k라고 하고 $f_k(x)$의 정의를 다시하면

$$f_k(x) = \sum_{j=k}^{\infty} \frac{x^j}{j!}, \qquad g(x) = \prod_{k=0}^{10} \{f_k(x)\}^{c_k}$$

가 되고 $\{f_k(x)\}^{c_k}$를 구하는 시간은 $O(L\lg L \cdot \lg c_k)$가 되어 총 시간은 $O(L\lg^2 L)$이 되어 빠르게 할 수 있다.

Stirling number 사용

아까와 같은 아이디어로 D_i가 같은 것들을 한꺼번에 처리한다.

동적계획법으로 잘 구할 수 있도록 조금 문제를 조금 바꿔서

“N개의 수를 K개의 그룹으로 나누는데 각 그룹에 r개 이상의 수가 있도록 하는 경우의 수”라고 하고 이를 $S_r(N,K)$라고 하면, 그룹을 구별할 수 있을 때의 경우의 수는 $K! \cdot S_R(N,K)$으로 이것이 우리가 구하는 수이다.

Stirling number 사용

$R = 1$이면 이를 Stirling number of the second kind라고 하고, $R > 1$이면 이를 Associated Stirling numbers of the second kind라고 한다.

점화식은 다음과 같다.

$$S_R(N,K) = KS_R(N,K) + \binom{N-1}{R-1} S_R(N-R, K-1)$$

Stirling number 사용

$$S_R(N,K) = KS_R(N-1,K) + \binom{N-1}{R-1} S_R(N-R,K-1)$$

1에서 N까지의 수로 그룹을 나누고, 구분을 위해 가장 그룹에서 가장 큰 수를 기억한다고 하자.

점화식의 의미

$$S_R(N,K) = \boxed{KS_R(N-1,K)} + \binom{N-1}{R-1} S_R(N-R,K-1)$$

$KS_R(N-1,K)$은 이미 있는 그룹에 N을 집어넣는다는 의미이다.

N이 있게 되는 그룹의 크기는 R보다 더 커지게 된다.

점화식의 의미

$$S_R(N,K) = KS_R(N-1,K) + \boxed{\binom{N-1}{R-1} S_R(N-R,K-1)}$$

$\binom{N-1}{R-1} S_R(N-R,K-1)$은 N을 위한 새로운 그룹을 만든다는 의미이다. N은 기본적으로 그룹에 포함되어 있으므로, 크기가 R인 그룹을 만들기 위해 다른 $R-1$개의 수를 고른다. 나머지 칸은 $K-1$개의 그룹을 만드는 경우의 수로 채울 수 있다.

Stirling number 사용

S_k는 K가 c_k가 될 때까지 구하면 된다. 즉, S_0에서 S_{10}까지를 구하는데 $O\big(N(c_0 + \cdots + c_{10})\big) = O(NL)$의 시간이 걸린다.

즉, 모든 $\{f_k(x)\}^{c_k}$를 구하는데 $O(NL)$의 시간이 걸리는 것이다.

그러므로 FFT가 필요없이 총 $O((N + L)L)$의 시간에 문제를 해결할 수 있다.

트리

정점이 N개인 Labeled tree중에서 특정한 간선 M개를 포함하는 것들의 개수를 구하여라.

Prüfer sequence

Labeled tree는 길이가 $N-2$이고, 각 원소가 정점의 번호인 어떤 수열과 일대일 대응이 된다. 이를 Prüfer sequence라고 하므로 처음 본다면 링크를 참조하도록 한다.

이 사실에 의해 정점의 개수가 N개인 서로 다른 Labeled tree의 개수는 N^{N-2}개가 된다.

특정 차수의 Labeled tree의 개수

어떤 Labeled tree를 Prüfer sequence로 바꾸고 나면, 이 수열이 생성되는 방식에 의해 어떤 정점이 이 수열에 나오는 횟수와 트리에서 이 정점의 차수에 1을 빼준 수가 같다.

즉, N개의 정점의 차수가 $d_1, \dots, d_N$인 트리의 개수는 아래와 같다.

$$\frac{(N-2)!}{(d_1-1)!\cdots(d_N-1)!}$$

어떤 간선이 들어 있어야 할 때

먼저 같은 컴포넌트에 속하는 간선들을 묶는다. 총 C개의 컴포넌트가 있어서, 각 컴포넌트의 크기를 $S_1, \dots, S_C$라고 하자.

각 컴포넌트를 슈퍼 노드로 만들고 각 컴포넌트의 차수가 $d_1, \dots, d_C$가 되도록 트리를 만든다면 만들 수 있는 총 개수는

어떤 간선이 들어 있어야 할 때

$$\sum_{\substack{d_1+\cdots+d_C = 2*(C-1) \\ 1 \le d_i \le C-1}} \binom{C-2}{d_1-1, \cdots, d_C-1} S_1^{d_1} \cdots S_C^{d_C}$$

이다. Multinomial은 특정한 차수에 대한 트리의 개수이고, 뒤에 곱해지는 S들은 간선이 특정 컴포넌트의 어떤 정점에 연결되어도 상관없기 때문에 곱해지는 수이다.

어떤 간선이 들어 있어야 할 때

$$S_1 \cdots S_C \left\{ \sum_{\substack{(d_1-1)+\cdots+(d_C-1)=C-2 \\ 0 \le d_i - 1 \le C-2}} \binom{C-2}{d_1-1, \cdots, d_C-1} S_1^{d_1-1} \cdots S_C^{d_C-1} \right\}$$

Multinomial theorem에 의해 이 값은

$$S_1 \cdots S_C (S_1 + \cdots + S_C)^{C-2} = S_1 \cdots S_C \, N^{C-2}$$

이다.

어떤 간선이 들어 있어야 할 때

컴포넌트는 합치는 것은 `disjoint-set` 자료 구조를 이용해서 할 수 있다. 앞에서 보이듯, 컴포넌트의 사이즈가 1인 것은 곱해줄 이유가 없으므로 처음부터 모든 정점을 만들지 말고 합쳐진 정점만 저장해 놓으면 된다.

어떤 컴포넌트에 간선 개수가 많아서 트리가 되지 않는 경우와 모든 정점이 하나의 컴포넌트에 들어가는 경우를 주의해야 한다.

팔찌

K종류의 구슬을 N개 이하만 사용하여 원형 팔찌를 만들 때, 만들 수 있는 종류의 개수를 구하여라.

단, 뒤집거나 돌려서 같은 구성이 되면 같은 종류이다.

Burnside's lemma

Burnside's lemma를 사용해야 할 것같은 문제이다.

이 문제에서 가능한 transform은 총 뒤집기와 돌리기의 조합에 의해 총 2N개가 있으며 각 경우를 살펴보도록 한다.

뒤집지 않고 돌리기만 하는 경우

먼저 각 구슬의 위치를 시계방향으로 0에서 $N-1$로 나타내면,

오른쪽으로 t만큼 돌렸을 때 x의 위치에 있는 구슬은 $(x+$

뒤집지 않고 돌리기만 하는 경우

$0 \le t < N$인 범위의 t에 대해 $\gcd(t, N) = \mathrm{d}$인 개수는

$\gcd\left(\frac{t}{d}, \frac{N}{d}\right) = 1$인 개수와 같으므로 $\varphi\left(\frac{N}{d}\right)$개 이다. $\varphi(x)$는 0이상 x미만의 정수 중에서 x와 서로소인 수의 개수이다.

이 연산에 대한 합은 다음과 같다.

$$\sum_{d|N} K^d \varphi\left(\frac{N}{d}\right)$$

한 번 뒤집고 돌리는 경우

한 번 뒤집으면 x의 위치에 있는 구슬은 $(-x) \bmod N$의 위치로 간다. 여기서 t만큼 오른쪽으로 뒤집으면 $(t-x) \bmod N$의 위치로 가는 것이다.

그러면 $(t-x) \bmod N$의 위치에 있는 구슬은 어디로 갈까? 바로 x로 가게 된다.

한 번 뒤집고 돌리는 경우

N이 홀수이면, $x \equiv t - x \pmod{N}$인 경우가 정확히 하나 존재하고, 나머지 수들이 $\frac{N-1}{2}$쌍의 짝을 짓게 되므로 같아지는 경우의 수가 하나의 t에 대해 $K^{\frac{N+1}{2}}$으로 총 $N \cdot K^{\frac{N+1}{2}}$이다.

한 번 뒤집고 돌리는 경우

N이 짝수이면, $x \equiv t - x (\mathrm{mod}\, N)$인 경우가 t가 홀수인 경우에는 없으므로 모든 수가 짝을 지어 같아지는 경우의 수가 $K^{\frac{N}{2}}$이 되고,

t가 짝수인 경우에는 $2x \equiv t\ (\mathrm{mod}\, N) \rightarrow x \equiv \frac{t}{2} \left(\mathrm{mod}\, \frac{N}{2}\right)$가 되어 두 가지 경우가 있어서 나머지 수들이 짝을 지으면 같아지는 경우의 수가 $K^{\frac{N}{2}+1}$이 되어 총합은 $\frac{N}{2}\left(K^{\frac{N}{2}} + K^{\frac{N}{2}+1}\right)$이 된다.

뒤집지 않고 돌리기만 하는 경우 총합

이제 이것을 $1 \le N \le n$인 범위에 대해 총합을 구해야 한다. 먼저 뒤집지 않고 돌리는 경우의 총합은

$$\sum_{N=1}^{n} \left\{ \frac{1}{2N} \sum_{d|N} K^d \, \varphi\left(\frac{N}{d}\right) \right\}$$

Burnsides’s lemma에서 합을 transform의 개수로 나누어 주어야 하기 때문에 중간에 $\frac{1}{2N}$이 들어갔음에 유의하라.

뒤집지 않고 돌리기만 하는 경우 총합

식을 조금 풀어서

$$\frac{1}{2}\sum_{N=1}^{n}\left\{\sum_{d|N}\frac{K^d}{N}\varphi\left(\frac{N}{d}\right)\right\}=\frac{1}{2}\sum_{N=1}^{n}\left\{\sum_{d|N}\frac{K^d}{d}\times\frac{\varphi\left(\frac{N}{d}\right)}{\frac{N}{d}}\right\}$$

뒤집지 않고 돌리기만 하는 경우 총합

$$\sum_{N=1}^{n}\left\{\sum_{d|N}\frac{K^d}{d}\times\frac{\varphi\left(\frac{N}{d}\right)}{\frac{N}{d}}\right\}$$

에서 d가 같을 때 $\frac{K^d}{d}$에 더해지는 수는 아래와 같다

$$\frac{\varphi(1)}{1}+\cdots+\frac{\varphi\left(\left\lfloor\frac{n}{d}\right\rfloor\right)}{\left\lfloor\frac{n}{d}\right\rfloor}$$

뒤집지 않고 돌리기만 하는 경우 총합

$\varphi(x)$를 구하는 데에 $O(nlgn)$

전체 합을 구하는 데에도 $O(nlgn)$의 시간이 걸린다.

한 번 뒤집고 돌리는 경우 총합

N이 홀수이면 $\frac{1}{2N}\left(N \cdot K^{\frac{N+1}{2}}\right) = \frac{1}{2}K^{\frac{N+1}{2}}$

N이 짝수이면 $\frac{1}{2N}\left\{\frac{N}{2}\left(K^{\frac{N}{2}} + K^{\frac{N}{2}+1}\right)\right\} = \frac{1}{4}\left(K^{\frac{N}{2}} + K^{\frac{N}{2}+1}\right)$

을 각 N에 대해 차례대로 더해주면 된다.

마지막으로 $N = 0$인 경우인 1을 더해주면 된다.

흑백

$H \times W$크기의 직사각형이 있고, 각 칸의 색은 50%의 확률로 흑백으로 정해진다.

몇 개의 열과 몇 개의 행을 선택하고 나머지 행열은 모두 없앨 때, 남은 칸이 모두 흑색이면 흑색 부분 직사각형, 백색이면 백색 부분 직사각형이라고 하자. 백색 부분 직사각형과 흑색 부분 직사각형 개수의 곱의 기댓값을 구하여라.

확률 변수로 나타내기

부분 직사각형으로 선택할 수 있는 경우의 수는 총

$$C = (2^H - 1)(2^W - 1)$$

가지, C개의 확률변수 $B_1, \cdots, B_C$가 서로 다른 부분 직사각형이 흑색 부분 직사각형이 되는 확률을 나타낸다면, 흑색 부분 직사각형의 개수를 나타내는 확률변수는 $B_1 + \cdots + B_C$가 될 것이다. 백색도 비슷하게 $W_1 + \cdots + W_C$로 나타낼 수 있다.

곱

우리가 구하는 것은 $E\big((B_1 + \cdots + B_C) \times (W_1 + \cdots + W_C)\big)$이다. 이를 전개해보면,

$$E\left(\sum_{i=1}^{C}\sum_{j=1}^{C} B_i W_j\right) = \sum_{i=1}^{C}\sum_{j=1}^{C} E\left(B_i W_j\right)$$

그러면 $E\left(B_i W_j\right)$를 모두 구해 더하면 된다.

경우

B_iW_j가 의미하는 바는 색이 다른 두 부분 직사각형이 한번에 나올 수 있느냐는 것이다. 다음의 세 가지 경우가 있다.

1) 두 부분 직사각형이 행과 열을 하나도 공유하지 않음

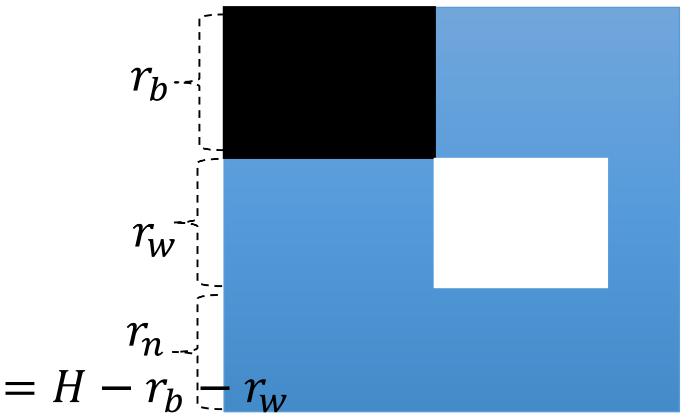
2) 두 부분 직사각형이 하나 이상의 행을 공유함

3) 두 부분 직사각형이 하나 이상의 열을 공유함

색이 다르기 때문에 행과 열을 동시에 공유할 수는 없다.

1번 경우

흑색 부분 직사각형이 차지하는 행,열을 r_b, c_b
백색 부분 직사각형이 차지하는 행,열을 r_w, c_w
아무도 차지하지 않는 행,열을 r_n, c_n이라 하고
가능한 모든 경우를 고려해보면



$$\sum_{\substack{1 \le r_b, r_w,\ 0 \le r_n \\ r_b + r_w + r_n = H}} \ \sum_{\substack{1 \le c_b, c_w,\ 0 \le c_n \\ c_b + c_w + c_n = W}} \binom{H}{r_b, r_w, r_n} \binom{W}{c_b, c_w, c_n} \left(\frac{1}{2}\right)^{r_b c_b + r_w c_w}$$

1번 경우

$$\sum_{\substack{1\le r_b, r_w,\ 0\le r_n \\ r_b+r_w+r_n=H}} \binom{H}{r_b, r_w, r_n} \left[\sum_{\substack{1\le c_b, c_w,\ 0\le c_n \\ c_b+c_w+c_n=W}} \binom{W}{c_b, c_w, c_n} \left\{ \left(\frac{1}{2^{r_b}}\right)^{c_b} \left(\frac{1}{2^{r_w}}\right)^{c_w} \right\} \right]$$

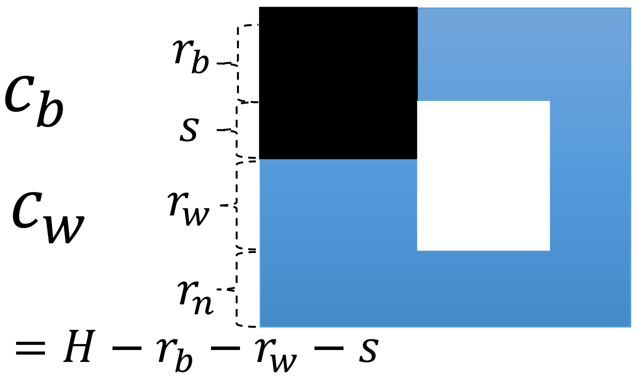
Multinomial theorem과 포함 배제의 원리를 이용하여

$$\sum_{\substack{1\le r_b, r_w,\ 0\le r_n \\ r_b+r_w+r_n=H}} \binom{H}{r_b, r_w, r_n} \left\{ \left(\frac{1}{2^{r_b}} + \frac{1}{2^{r_w}} + 1\right)^W - \left(\frac{1}{2^{r_b}} + 1\right)^W - \left(\frac{1}{2^{r_w}} + 1\right)^W + 1 \right\}$$

이 값은 직접 r_b, r_w에 대한 for문을 돌아 구할 수 있다.

2번 경우(행 공유)

흑색 부분 직사각형이 홀로 차지하는 행,열을 r_b, c_b
백색 부분 직사각형이 홀로 차지하는 행,열을 r_w, c_w
아무도 차지하지 않는 행,열을 r_n, c_n
공유되고 있는 행의 개수를 s라고 하면



$$\sum_{\substack{1\le s,\ 0\le r_b, r_w, r_n \\ r_b+r_w+r_n+s=H}} \ \sum_{\substack{1\le c_b, c_w,\ 0\le c_n \\ c_b+c_w+c_n=W}} \binom{H}{r_b, r_w, r_n, s}\binom{W}{c_b, c_w, c_n}\left(\frac{1}{2}\right)^{(r_b+s)c_b+(r_w+s)c_w}$$

2번 경우(행 공유)

$$\sum_{\substack{1\le c_b, c_w,\ 0\le c_n \\ c_b+c_w+c_n=W}} \binom{W}{c_b, c_w, c_n} \sum_{\substack{1\le s,\ 0\le r_b, r_w, r_n \\ r_b+r_w+r_n+s=H}} \binom{H}{r_b, r_w, r_n, s} \left\{ \left(\frac{1}{2^{c_b}}\right)^{r_b} \left(\frac{1}{2^{c_w}}\right)^{r_w} \left(\frac{1}{2^{c_b+c_w}}\right)^{s} \right\}$$

Multinomial theorem과 포함 배제의 원리를 이용하여

$$\sum_{\substack{1\le c_b, c_w,\ 0\le c_n \\ c_b+c_w+c_n=W}} \binom{W}{c_b, c_w, c_n} \left\{ \left(\frac{1}{2^{c_b}} + \frac{1}{2^{c_w}} + \frac{1}{2^{c_b+c_w}} + 1\right)^{H} - \left(\frac{1}{2^{c_b}} + \frac{1}{2^{c_w}} + 1\right)^{H} \right\}$$

이 값은 직접 c_b, c_w에 대한 for문을 돌아 구할 수 있다.

3번 경우(열 공유)

2번 경우를 H, W를 바꾸어 하는 것과 같다.