SCPC
예수시피시

05.21(일) 14:00~17:00

object 'SCPC' has no attribute 'div'

해설 17:00~19:00

A-미소녀 컴퓨터 파루빗토 쨩

최초 풀이자: asoek46521(14분)

A-미소녀 컴퓨터 파루빗토 쨩

- 가능한 모든 입력을 생각해보면, 입력된 문자열 s에 대해 0번째 인덱스를 제외한 위치에서 가장 먼저 나오는 문자를 기준으로 파싱이 가능함을 알 수 있습니다.
- 이를 이용하면 A,B, 연산자를 간단하게 분리해낼 수 있습니다.

A-미소녀 컴퓨터 파루빗토 짱

- 분리를 해냈으면 실제로 계산해보면 됩니다. Floor연산임에 유의하세요.

- Cpp의 stringstream메소드나, python의 int, oct 메소드를 사용하면 쉽게 구현할 수 있습니다.

B-고인물의 졸업 계획

최초 풀이자: bnb2011(20분)

B-고인물의 졸업 계획

- $N \le S_i \le 2N$을 만족하는 S_i가 존재한다면 그 1과목만 출력하면 됩니다.

B-고인물의 졸업 계획

- $N \le S_i \le 2N$을 만족하는 S_i가 존재하지 않는다고 가정합니다. 문제 조건에 의해 $N \le \sum_{k=1}^{L} S_{X_k} \le 2N$를 만족하는 집합이 반드시 존재합니다.

- $S_i < N$인 S_i를 찾을 때마다 계속해서 더하는 전략을 생각합니다.

B-고인물의 졸업 계획

- 만약 이 전략으로 순회를 하였음에도 불구하고 합이 N이상이 되지 못했다면 문제 조건에 모순입니다.

- 만약 이 전략으로 순회를 하였음에도 불구하고 합이 $N \le \sum_{k=1}^{L} S_{X_k} \le 2N$범위 안에 들어가지 못하고 바로 $2N$을 초과해버린다면 $S_i < N$이라는 조건에 모순입니다.

- 따라서 그리디하게 문제를 해결할 수 있습니다.

C-응애(EASY)

최초 풀이자: seawon0808(3분)

C-응애(EASY)

- i번째 인사 상태를 S_i라고 할 때, S_{i+1}은 S_i에 의해서만 결정됩니다.

- S_{i+1}에서 k번째 학생의 상태는 S_i에서 $k-1$번째 학생과 $k+1$번째 학생의 상태를 xor한 결과에 의해 결정됩니다.

- N, K의 범위가 작으므로 naïve하게 xor을 구현해도 AC를 받을 수 있습니다.

D-응애(HARD)

최초 풀이자:hjroh0315(42분)

D-응애(HARD)

- S_{i+1}은 S_i에 의해서만 결정된다는 부분에 집중해봅시다.
- $S_{i+1} = \mathrm{f}(S_i)$라고 하면, f는 Z_2상에서 S_i에 대한 선형변환임을 알 수 있습니다.
- $\mathrm{f}(S_i) = w \times S_i$라고 할 때 w를 구해 봅시다.

D-응애(HARD)

- $\mathrm{N} = 6$일 때 w를 구해 보면 다음과 같습니다.
- $w = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$
- 이를 A와 B로 쪼갤 수 있습니다.

D-응애(HARD)

- $A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
- $B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$
- 이때 $w = A + B$입니다.

D-응애(HARD)

- $w \times S_i = (A + B)S_i = A \times S_i + B \times S_i$에서 A와 B는 모두 S_i에 대한 cyclic shift입니다.
- 초기상태 S_0에 K번 연산을 진행한 후의 상태는 $w^K \times S_0$입니다.
- $AB = I$입니다.
- 따라서 $w^2 = A^2 + 2I + B^2 = A^2 + B^2$입니다.

D-응애(HARD)

- 이를 반복하면 $w^{2^j} = A^{2^j} + B^{2^j}$가 성립한다는 사실을 알 수 있습니다.
- $K = 13$일때를 예로 들면 $13 = 8 + 4 + 1$이므로 8번, 4번, 1번 shift한 후 xor하는 것을 반복하면 됩니다.
- 시간복잡도는 $\mathrm{O(NlogK)}$입니다.

E- 특별한 한붓그리기

최초 풀이자:menborong(75분)

E- 특별한 한붓그리기

- $A \to B$로의 임의의 간선을 긋고, 다시 $B \to A$로의 간선을 긋는 것을 반복한다고 생각해봅시다.

- 만약 $B \to A$로의 간선을 긋는 것이 불가능해지면 뉴비가 승리하게 됩니다. 이외에는 고인물이 승리하게 됩니다.

- 그렇다면 이분 매칭으로 문제를 해결할 수 있습니다. A의 노드의 개수가 450으로 크지 않기 때문에 A의 노드에서 매칭을 시작하여 문제를 해결할 수 있습니다.

E- 특별한 한붓그리기

- A에서 매칭을 시도했을 때 실패한다면 이는 $B \rightarrow A$로의 간선을 그을 수 없게 되었다는 뜻이므로 뉴비가 승리함을 의미합니다.

- 그 외에는 항상 고인물이 승리하게 됩니다.

F- 사각형 분할

최초 풀이자: -

F- 사각형 분할

다음 내용은 증명으로, 알고리즘만 보시려면 29슬라이드로 넘어가셔도 됩니다.

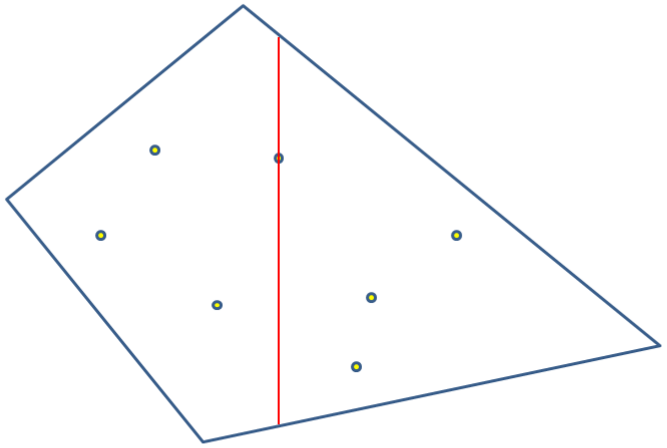
임의의 볼록 사각형 R이 존재하고, $S \cup S'$원소 중 R내부에 포함되는 점의 집합을 S''이라고 합시다.

F- 사각형 분할

S''중 임의의 점 P를 고르고, P만 지나는 직선 l을 다음 조건을 만족하도록 고릅니다.

- 직선 l이 R을 다각형 A', B'으로 분할하고, A', B' 내부의 점의 개수의 차가 1이하이다.

위와 같은 상태를 "초기 상태"라 정의합니다.

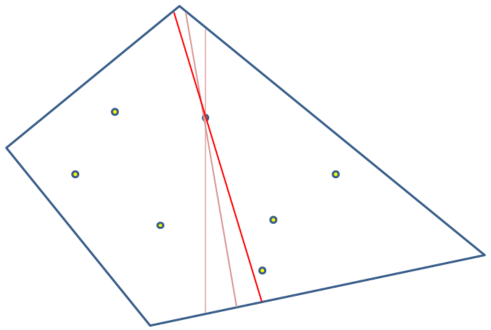


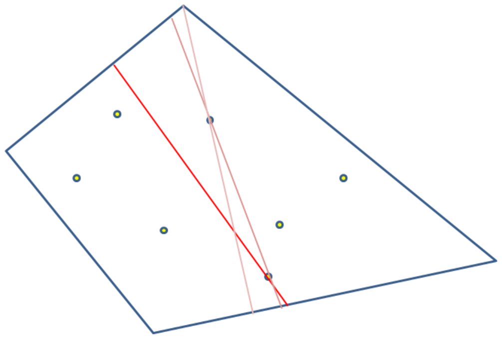
F- 사각형 분할

다음 과정은 “회전”이라 정의합니다.

점 P를 중심으로 점 P외의 다른 점을 만나기 전까지, 직선 l을 반시계 방향으로 회전시킨다.

만약, 직선 l이 점 P외 다른 점을 지나면, 그 점을 새 P로 잡는다.





F- 사각형 분할

회전 과정을 반복하는 동안, 직선 l이 두 점을 지나지 않는 경우에는, 분할한 두 다각형의 점의 개수는 보존됩니다.

이는 회전 과정에서 점 P를 바꿀 때, 바꾸기 전의 P점과 바꾼 후의 P점이 같은 다각형 내에 있기 때문입니다.

F- 사각형 분할

위 과정을 직선 l이 180° 회전할 때까지 반복하면, n(S'')의 기우성에 따라 2가지 결과가 나옵니다.

홀수인 경우는 다시 초기 상태로 돌아옵니다.

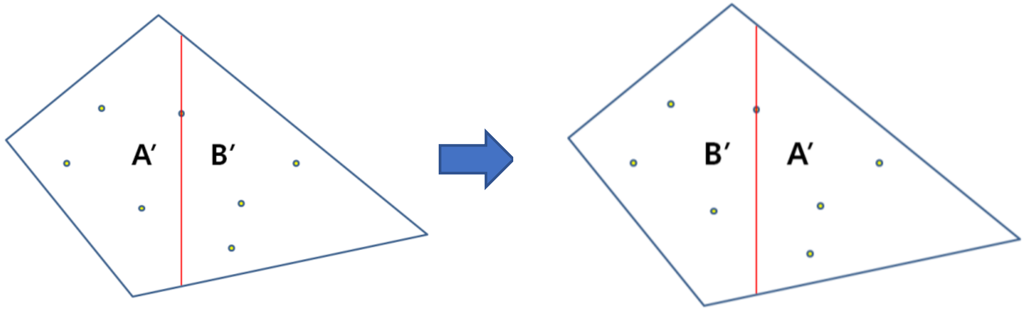
짝수인 경우에는, 처음의 점과 다른 점을 P로 갖습니다.

F- 사각형 분할

홀수인 경우는, 결국 다각형 A', B'이 서로 바뀐 경우입니다.

넓이는 연속적으로 변화하므로, 사이값 정리에 따라 직선 l이 볼록 사각형을 이등분하는 경우가 존재합니다.

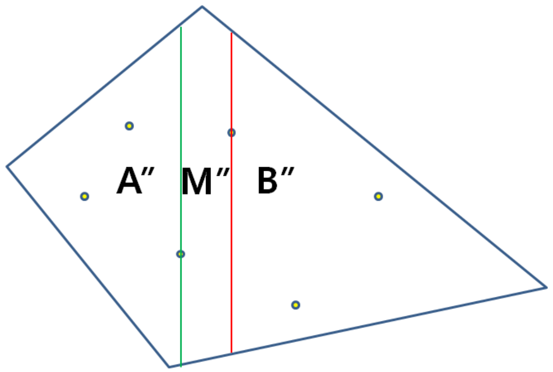
따라서 상한은 $\left(\frac{(R\text{의 넓이})}{2}\right)^2$ 입니다.



F- 사각형 분할

짝수의 경우에는 초기 상태의 직선을 l_0, 180° 회전 후 상태의 직선을 l_1이라 합시다.

l_0과 R의 경계로 이루어진 다각형을 A'', l_1과 R의 경계로 이루어진 다각형을 B'', l_0,l_1과 R의 경계로 이루어진 다각형을 M'' 이라 합시다. (세 다각형은 겹치지 않습니다.)



F- 사각형 분할

만약, A'', B$''$ 모두 R의 절반보다 넓이가 작다면, M'' 내부에 임의의 직선을 그어서 넓이를 이등분할 수 있습니다.

그렇지 않다면, 사이값 정리에 따라 직선 l이 넓이를 이등분하는 경우가 존재합니다.

따라서 상한은 $(\frac{(R\text{의 넓이})}{2})^2$입니다.

F- 사각형 분할

따라서, (A의 넓이) * (B의 넓이) 의 상한은, 볼록 사각형 최대의 넓이 절반의 제곱입니다.

결국 볼록 사각형 넓이의 최대값을 구하는 문제가 됩니다.

Convex Hull을 먼저 구합시다.

F- 사각형 분할

Convex Hull 을 구성하는 정점 중, 사각형의 대각선을 이룰 정점 하나를 먼저 고릅니다.

대각선을 이룰 나머지 정점은, 처음 고른 정점과 이웃한 정점으로부터 시작하여, 반시계 방향으로 회전시켜가며 고를 수 있습니다.

F- 사각형 분할

대각선을 이루는 정점 2개를 고르면, 나머지 두 점은 대각선으로부터 가장 멀리 떨어진 두 점을 고르면 됩니다.

이때, 멀리 떨어진 점을 찾는 과정에서 대각선이 반시계 방향으로 회전한다는 특징을 이용하면, 결국 가장 멀리 떨어진 점들도 반시계 방향으로 회전함을 알 수 있습니다.

F- 사각형 분할

대각선이 회전할 때마다, 가장 멀리 떨어진 점들을 반시계 방향으로 탐색하며, 갱신합니다.

이러한 사실들을 종합하면, $O(N^2)$ 시간 내에 가장 큰 사각형의 넓이를 찾을 수 있습니다.

마지막으로 최대값의 절반의 제곱을 출력하면 됩니다.

G- 우선순위와 쿼리

최초 풀이자:heeda0528(34분)

G- 우선순위와 쿼리

- 이 문제에서 쿼리는 2가지입니다.
- 하나는 정점을 잇는 쿼리, 다른 하나는 가장 정점의 개수가 많은 트리를 삭제하는 쿼리입니다.
- 정점간 연결된 상태를 저장하기 위해 Disjoint set을 이용합니다.
- 정점을 연결할 때마다, Union 연산을 수행합니다.

G- 우선순위와 쿼리

- 합칠 때 우선순위를 고려하여, 번호가 더 작은 정점을 Root로 둡니다.

- 만약, Union 연산을 실행할 때, 두 정점이 같은 집합에 속한다면, 이는 Cycle을 형성하는 것을 의미합니다.

- Cycle을 형성하는 그래프는 트리가 아니기에, 해당 그래프가 트리인지 체크하는 배열이 필요합니다.

G- 우선순위와 쿼리

- 이 배열의 값이 바뀌는 경우는 다음 2가지 입니다

- 1. 같은 그래프 내의 두 정점을 이어서 Cycle을 형성하는 경우.
- 2. 트리인 그래프와 트리가 아닌 그래프를 잇는 경우.

- 또한, 그래프의 크기를 저장하는 배열을 만들어서, Union 연산을 할 때마다 갱신합니다.

G- 우선순위와 쿼리

- 두 번째 쿼리는 가장 정점의 개수가 많은 트리를 출력하는 것입니다.
- 이 때 만약 전체 트리를 선형적으로 탐색한다면, 시간복잡도가 O(N)이므로, 시간을 초과할 수 있습니다.
- 따라서, O(logN) 시간 안에 삽입, 검색, 삭제가 가능한 레드-블랙 트리 자료구조를 사용합니다.

G- 우선순위와 쿼리

- 레드-블랙 트리에는 각 트리 안에서 가장 번호가 작은 정점과 트리의 크기를 Pair로 묶어서 저장합니다.
- 그리고, 우선순위를 크기가 클수록, 크기가 같다면, 번호가 낮을수록 높게 지정합니다.
- 처음에 레드-블랙 트리에는 모든 정점이 들어있습니다.

G- 우선순위와 쿼리

- 정점을 연결할 때는, 각 정점에서 find 연산으로 root를 찾고, 이를 key값으로 레드-블랙 트리에서 찾고, 삭제합니다.

- 이 때 두 그래프를 합친 후, 그래프가 트리라면, 다시 레드-블랙 트리에 삽입합니다.

- 이 과정을 통해, 레드-블랙 트리에는 트리인 그래프만 들어있게 됩니다.

G- 우선순위와 쿼리

- 마지막으로, 2번 쿼리를 실행할 때는, 레드-블랙 트리에서 가장 우선순위가 높은 값을 찾아서 출력하고 삭제합니다.

- 이외에도 레드-블랙 트리 대신에 indexed heap이나 B-tree를 쓸 수 있고, 일반적인 heap을 2개 이용하여 풀 수도 있습니다.

EX-비밀의 채팅방

최초 풀이자: -

EX-비밀의 채팅방

- 각 단어에 담긴 비밀의 힘이 $\{1,2,\ldots,M^N\}$입니다. 이를 다항식으로 표현해봅시다.

- $f(x) = (x+1)(x^2+1)\ldots\left(x^{M^N}+1\right)$을 생각하면, 가능한 모든 문장을 다항식 $f(x)$로 표현할 수 있다는 것을 알 수 있습니다. 이때 $f(x)$의 각 항을 kx^s라는 것은 mod를 취하기 전 비밀의 힘이 s가 되는 문장의 개수가 k라는 사실을 의미합니다.

EX-비밀의 채팅방

- mod(M)을 하는 상황이므로 복소해석의 아이디어를 잠깐 빌려오도록 합시다.

- $\delta_1 = e^{1\times\frac{2\pi i}{M}}$, $\delta_2 = e^{2\times\frac{2\pi i}{M}}, \ldots, \delta_M = e^{M\times\frac{2\pi i}{M}}$를 각각 $f(x)$에 대입해보면 mod(M)을 했을 때 0이 되는 항만 골라낼 수 있습니다. (복소평면에서의 무게중심)

EX-비밀의 채팅방

- $M = 6$인 상황을 예시로 들어보겠습니다.
- $\delta_1 = e^{1\times\frac{2\pi i}{6}}$, $\delta_2 = \ e^{2\times\frac{2\pi i}{6}}, \ldots, \ \delta_6 = e^{6\times\frac{2\pi i}{6}}$입니다.
- $f(x) = 1 + c_1x + c_2x^2 + \cdots c_{M^N}x^{M^N}$입니다.
- $\sum_{k=1}^{6} f(\delta_k)$을 생각해봅시다.

EX-비밀의 채팅방

- 우선 정의에 의해, $\delta_n = e^{n\times\frac{2\pi i}{6}}$이므로 $\delta_n^6 = 1$입니다.(주기가 6)
- 각 항별로 덧셈을 진행하면,

  상수항의 덧셈 결과는 6입니다.
  x^1항의 덧셈 결과는 $c_1 \sum_{k=1}^{6} \delta_k = 0$ 입니다.
  x^2항의 덧셈 결과는 $c_2 \sum_{k=1}^{6} \delta_k^2 = 0$ 입니다.
  x^3항의 덧셈 결과는 $c_3 \sum_{k=1}^{6} \delta_k^3 = 0$ 입니다.
  x^4항의 덧셈 결과는 $c_4 \sum_{k=1}^{6} \delta_k^4 = 0$ 입니다.
  x^5항의 덧셈 결과는 $c_5 \sum_{k=1}^{6} \delta_k^5 = 0$ 입니다.
  x^6항의 덧셈 결과는 $c_6 \sum_{k=1}^{6} \delta_k^6 = 6c_6$ 입니다.

EX-비밀의 채팅방

- 이를 복소평면에 표시해보면 x의 지수가 6의 배수인 경우만 골라낼 수 있다는 사실을 알 수 있습니다.

- 결국 정리하면, 이 문제는 $\sum_{k=1}^{M} f(\delta_k)$ 을 구하는 문제라는 것을 알 수 있습니다. 이를 구할 수 있는 간단한 아이디어는 다음과 같습니다.

EX-비밀의 채팅방

- 제일 처음에 $\delta_n = e^{n \times \frac{2\pi i}{M}}$로 정의했다는 점을 떠올려봅시다.

- $\delta_n^M = 1$이므로, 간단하게
  $1 - x^M = (\delta_1 - x)(\delta_2 - x) \ldots (\delta_M - x)$입니다.

- 위 식의 x에 -1을 대입하면 $(\delta_1 + 1)(\delta_2 + 1) \ldots (\delta_M + 1)$의 값은 M이 짝수인 경우 0, M이 홀수인 경우 2가 됨을 알 수 있습니다.

EX-비밀의 채팅방

- $\delta_n^M = 1$이므로, $f(x) = (x+1)(x^2+1)\dots\left(x^{M^N}+1\right)$을 생각하면 $f(\delta_1) = \left((\delta_1+1)(\delta_2+1)\dots(\delta_M+1)\right)^{M^{N-1}}$입니다.

- 이제 뭔가 좀 보이기 시작합니다.

EX-비밀의 채팅방

- $M = 6$인 예시로 다시 돌아가봅시다.
- $f(\delta_1) = 0$임은 이제 쉽게 알 수 있습니다.
- $f(\delta_2)$을 생각해보면

$$f(\delta_2) = \left(\left(\delta_1^2 + 1\right)\left(\delta_2^2 + 1\right) \dots \left(\delta_6^2 + 1\right)\right)^{6^{N-1}}$$

$$= \left((\delta_2 + 1)(\delta_4 + 1)(\delta_6 + 1)(\delta_2 + 1)(\delta_4 + 1)(\delta_6 + 1)\right)^{6^{N-1}}$$

$= \left((\delta_2 + 1)(\delta_4 + 1)(\delta_6 + 1)\right)^{2\times 6^{N-1}}$입니다.

EX-비밀의 채팅방

- 앞에서 했던 증명을 다시 가져와봅시다.

- $\delta_2^{\frac{M}{2}} = 1$이므로, 간단하게
  $1 - x^{\frac{M}{2}} = (\delta_2 - x)(\delta_4 - x)(\delta_6 - x)$입니다.

- 위 식의 x에 -1을 대입하면 $(\delta_1 + 1)(\delta_2 + 1) \ldots (\delta_M + 1)$의 값은 2가 됨을 알 수 있습니다.

EX-비밀의 채팅방

- 이걸 반복하여 0,2를 계속해서 판별해주며 더하면 이제 비밀번호를 구할 수 있습니다!
- 이제 비밀번호를 구하는 방법은 알았습니다. 그럼 이걸 어떻게 빠르게 구할 수 있을까요?

EX-비밀의 채팅방

- $f(x)$에 각각 δ_n을 넣었을 때 사이클이 어떻게 변하는지 관찰해봅시다.
- 어차피 $\delta_n^{\mathrm{M}} = 1$이니까 그만큼만 관찰해도 됩니다.
- $M = 6$인 예시를 다시 가져와봅시다.

EX-비밀의 채팅방

$f(\delta_1)$에서의 사이클은 $\{\delta_6, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5\}$입니다.
$f(\delta_2)$에서의 사이클은 $\{\delta_6, \delta_2, \delta_4, \delta_6, \delta_2, \delta_4\}$입니다.
$f(\delta_3)$에서의 사이클은 $\{\delta_6, \delta_3, \delta_6, \delta_3, \delta_6, \delta_3\}$입니다.
$f(\delta_4)$에서의 사이클은 $\{\delta_6, \delta_4, \delta_2, \delta_6, \delta_4, \delta_2\}$입니다.
$f(\delta_5)$에서의 사이클은 $\{\delta_6, \delta_5, \delta_4, \delta_3, \delta_2, \delta_1\}$입니다.
$f(\delta_6)$에서의 사이클은 $\{\delta_6, \delta_6, \delta_6, \delta_6, \delta_6, \delta_6\}$입니다.
하나의 주기에 대해서 사이클의 크기를 dict로 표현하면
$\{1{:}\,1, 2{:}\,1, 3{:}\,2, 6{:}\,2\}$입니다. 이때 dict에서 key는 사이클의 크기, value는 하나의 주기에 대해서 그 사이클이 발생하는 횟수입니다.

EX-비밀의 채팅방

- 앞 슬라이드에서 표현한 방식으로 16을 표현하면 $\{1:1, 2:1, 4:2, 8:4, 16:8\}$입니다.

- 앞 슬라이드에서 표현한 방식으로 81을 표현하면 $\{1:1, 3:2, 9:6, 27:18, 81:54\}$입니다.

- 일반화하면, 소인수 p의 N제곱 형태의 숫자는
$$\{\mathrm{p^i: (p-1) * p^{i-1}\ if\ i! = 0\ else\ 1\ for\ i\ in\ range(N+1)}\}$$
의 형태로 표현할 수 있습니다. 이는 $\varphi(n)$를 이용해서 증명할 수 있습니다.

EX-비밀의 채팅방

- 또한, 16*81을 이 방식으로 표현하면 $\{1:1, 2:1, 3:2, 4:2, 6:2, 8:4, 9:6, 12:4, 16:8, 18:6, 24:8, 27:18, 36:12, 48:16, 54:18, 72:24, 81:54, 108:36, 144:48, 162:54, 216:72, 324:108, 432:144, 648:216, 1296:432\}$입니다.

- 눈치채셨나요? A*B의 dict 표현은 element-wise한 곱셈으로 표현 가능합니다.

EX-비밀의 채팅방

- 이를 코드로 표현하면
  $\text{AmultiB} = \{\}$
  for i in dictA:
  $\quad \text{for j in dictB:}$
  $\qquad \text{AmultiB[i * j] = AmultiB.get(i * j, 0) + dictA[i] * dictB[j]}$
  입니다.
- 이런 표현이 가능한 이유는 마찬가지로 $\varphi(n)$의 성질 때문입니다.
- $n = p^r q^s$에 대해서 $n = \sum_{i=0}^{r} \sum_{j=0}^{s} \varphi(\frac{n}{p^i q^j})$가 성립합니다.

EX-비밀의 채팅방

- 이제 다 왔습니다. 임의의 정수 M에 대해 소인수분해 한 후 이를 모두 곱하면 M의 사이클을 분석한 dict를 얻을 수 있습니다. 이를 이용해서 $\sum_{k=1}^{M} f(\delta_k)$ 를 계산한 후 mod를 취해주면 됩니다.

- 너무도 자명하게, 문제에서 구하라고 한 확률 p_M/q_M에서, $p_M = \frac{\sum_{k=1}^{M} f(\delta_k)}{M}$ 이고, $q_M = 2^{M^N}$입니다.

EX-비밀의 채팅방

- 여기까지 도착하셨다면 다들 아실 거라고 생각하지만, $a^{p-1} \equiv 1 \pmod p$에서 $2^{M^N} \equiv 2^{M^{\mathrm{N}} \% (10^9+6)}$이므로 이를 $\mathrm{pow}(2, \mathrm{pow}(\mathrm{M}, \mathrm{N}, 10^9+6), 10^9+7)$로 빠르게 구할 수 있습니다.

- 이제 임의의 M에 대해 비밀번호를 충분히 빠르게 구할 수 있으므로 완전 탐색으로 문제를 해결할 수 있습니다.

EX-비밀의 채팅방

- 소인수분해를 naïve하게 하면 TLE를 받을 수 있습니다. 폴라드-로 등의 방법을 사용하면 코드를 더 빠르게 돌릴 수 있습니다.

- 범위가 작기 때문에 밀러-라빈보다는 에라토스테네스의 체를 전처리해서 소수 판정을 하는게 더 빠를 수 있습니다.

- 다들 $\bmod M$하면 $\% M$만 생각하시길래 그렇지 않은 문제를 준비했습니다.