2024 Goricon 풀이

by

Official Solutions

문제		의도한 난이도	출제자
A	문자열을 만들어요	**Very Easy**	hamuim
B	등차수열을 만들어요	**Easy**	hamuim
C	게임을 만들어요	**Medium**	hamuim
D	정다각형를 만들어요	**Medium**	hamuim
E	세계를 만들어요	**Medium**	hamuim
F	최솟값을 만들어요	**Hard**	hamuim
G	직각삼각형을 만들어요	**Hard**	hamuim
H	나머지를 만들어요	**Challenging**	hamuim

A. 문자열을 만들어요

`string`
출제진 의도 – **Very Easy**

- 출제자: `hamuim`

A. 문자열을 만들어요

- ✓ 길이가 4인 모든 연속 부분 문자열에 대해 `gori`와 같은지 확인하면 됩니다.
- ✓ 시간복잡도는 $\mathcal{O}(N)$입니다.

B. 등차수열을 만들어요

`ad_hoc`, `math`
출제진 의도 – **Easy**

- 출제자: `hamuim`

B. 등차수열을 만들어요

- M 이 1 보다 크기 때문에 M 에 2 이상의 양의 정수를 곱하게 되면 항상 합성수가 되어 소수가 아니게 됩니다.
- M 에 2 부터 $N+1$ 까지 곱한수를 차례대로 나열해 등차수열을 완성할 수 있습니다.
- 시간복잡도는 $\mathcal{O}(N)$ 입니다.

C. 게임을 만들어요

`ad_hoc`, `game_theory`
출제진 의도 – **Medium**

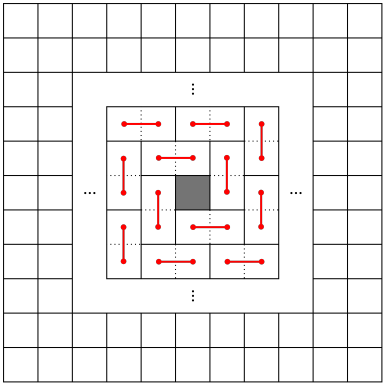
- 출제자: `hamuim`

C. 게임을 만들어요

- ✓ N이 3일 때는 선공이 항상 첫 번째 턴에 말을 테두리로 이동시킬 수 있기 때문에 이기게 됩니다.
- ✓ N이 5일 때는 후공이 항상 두 번째 턴에 말을 테두리로 이동시킬 수 있기 때문에 이기게 됩니다.
- ✓ $N \geq 7$인 경우만 생각하겠습니다.

C. 게임을 만들어요

- 격자의 테두리와 격자의 테두리에 인접한 칸을 제외하고 그림과 같이 2칸씩 묶어줍니다.



C. 게임을 만들어요

- 이제 후공은 선공이 격자의 어떤 칸에 도달하든 2칸으로 묶여있는 다른 칸으로 이동하는 전략을 사용할 겁니다.
- 위 전략을 사용하면 선공이 먼저 말을 움직일 수 없게되거나 후공이 먼저 격자의 테두리에 인접한 칸으로 도달해 격자의 테두리로 말을 이동하게 됩니다.
- 그러므로 $N \geq 7$인 경우에는 항상 후공이 이기게 됩니다.
- 시간복잡도는 $\mathcal{O}(1)$입니다.

D. 정다각형을 만들어요

`combinatorics`, `graph_traversal`
출제진 의도 – **Medium**

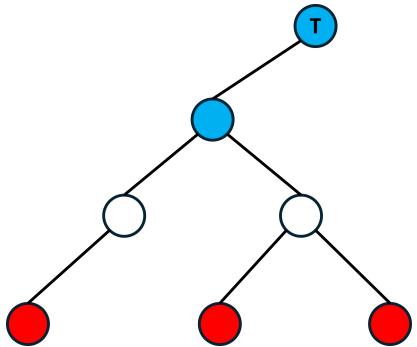
- 출제자: `hamuim`

D. 정다각형을 만들어요

- 정다각형의 정점들과 거리가 같은 정점 T를 먼저 정한 다음 T를 트리의 루트라 생각하고 문제를 풀어봅시다.
- T와 연결된 간선의 개수가 하나라고 생각해 보겠습니다.

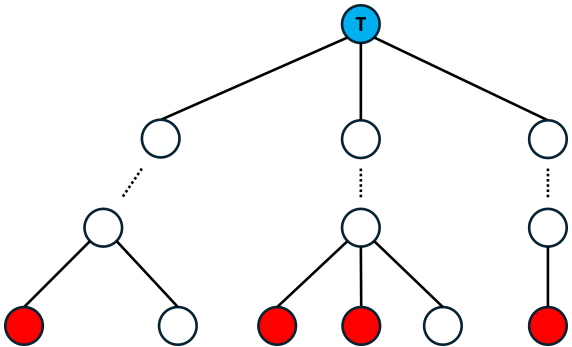
D. 정다각형을 만들어요

- 아래 그림과 같이 정다각형을 만들게되면 T와 간선으로 연결된 정점 또한 거리가 같게 되므로 T에 연결된 간선의 개수가 하나인 경우에는 정다각형을 만들 수 없습니다.



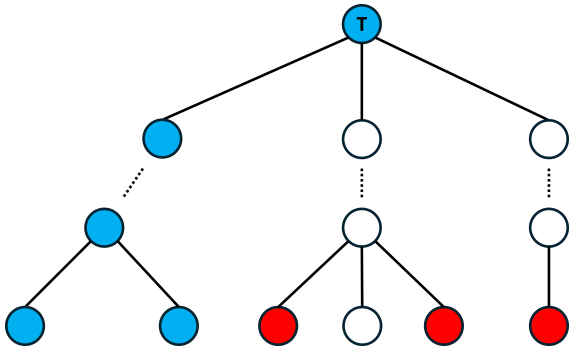
D. 정다각형을 만들어요

- ✓ T와 연결된 간선의 개수가 둘 이상일 때를 보겠습니다.
- ✓ 아래 그림과 같이 정다각형을 만들게 되면 T를 제외하고는 거리가 같은 정점이 존재하지 않습니다.



D. 정다각형을 만들어요

- 하지만 아래 그림과 같이 정다각형을 만들게 되면 T 와 간선으로 연결된 한 정점을 루트로 하는 서브 트리의 모든 정점이 정다각형과 거리가 같아지게 됩니다.



D. 정다각형을 만들어요

- 이를 통해 T와 간선으로 연결된 정점들을 루트로 하는 서브 트리 모두에 적어도 하나의 정점이 존재해야 정다각형을 만들 수 있다는 것을 알 수 있습니다.
- T와 간선으로 연결된 정점의 개수가 i일 때 T와 간선으로 연결된 정점들을 루트로 하는 서브 트리에서 각 서브 트리마다 T와의 거리가 X인 정점의 개수를 $x_{1,X}, x_{2,X}, \cdots, x_{i,X}$라 하겠습니다.
- 모든 서브 트리에 정점이 적어도 하나는 존재해야 하므로 T와의 거리가 X인 정다각형의 개수는 $(2^{x_{1,X}}-1)(2^{x_{2,X}}-1)\cdots(2^{x_{i,X}}-1)$이 됩니다.

D. 정다각형을 만들어요

- T에서 시작해 너비 우선 탐색을 하여 각 서브 트리마다 T와의 거리가 같은 정점의 개수를 구할 수 있고 이들로부터 정다각형의 개수를 계산할 수 있습니다.
- 두 정점 사이의 거리는 N보다 작기 때문에 $2^0, 2^1 \cdots 2^N$ 의 $\mod 10^9 + 7$을 한 값을 미리 계산해 줄 수 있습니다.
- 시간복잡도는 $\mathcal{O}\left(N^2\right)$ 입니다.

E. 세계를 만들어요

`constructive`, `graph_theory`
출제진 의도 – **Medium**

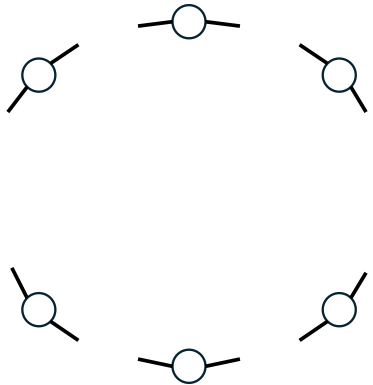
- 출제자: `hamuim`

E. 세계를 만들어요

- ✓ $N = 1$인 경우에는 $M = 1$이면 모든 정점끼리 연결해 주고 $M = 2$이면 3개의 정점에 6개의 간선을 연결할 수 없기 때문에 답은 NO입니다.
- ✓ $N \geq 2$라면 먼저 i번 정점에서 $i + 1$번 정점으로 N번 정점은 1번 정점과 간선을 이어 원형이며 모든 정점이 연결된 그래프로 만듭니다.
- ✓ 각각의 정점은 2개의 간선과 연결되어 있고 방금 $3N$개의 간선을 사용했기에 남은 간선의 개수는 $3(M - N)$개입니다.
- ✓ 정점을 6개씩 보면서 남은 간선들도 이어봅시다.

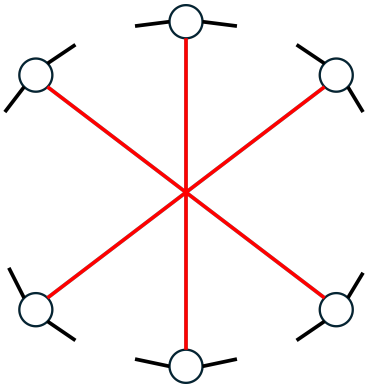
E. 세계를 만들어요

✓ 연결된 간선이 2개인 정점 6개를 선택하면 아래 그림과 같습니다.



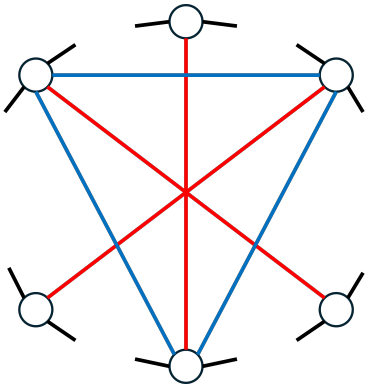
E. 세계를 만들어요

- 아래 그림과 같이 3개의 간선을 이어 6개의 정점이 3개의 간선과 연결되게 할 수 있습니다.



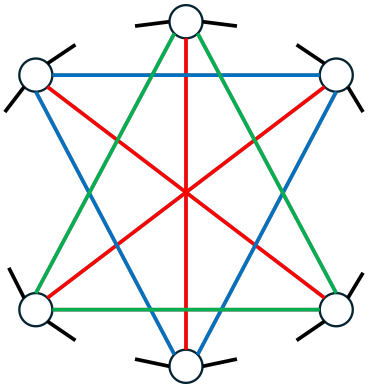
E. 세계를 만들어요

- 추가로 아래 그림과 같이 3 개의 간선을 이어 3 개의 정점만 5 개의 간선과 연결되게 할 수 있습니다.



E. 세계를 만들어요

- 마지막으로 아래 그림과 같이 3개의 간선을 이어 6개의 정점이 5개의 간선과 연결되게 할 수 있습니다.



E. 세계를 만들어요

- 결국 6 개의 정점을 선택해 $3+3+3=9$ 개의 간선을 이어 줄 수 있기에 미리 이어놓은 $3N$ 개의 간선까지 생각하면 위 방법으로 그래프에서 최대 약 $7.5N$ 개의 간선을 이어줄 수 있으며 $3(M-N)$ 은 3 의 배수이고 $3N \leq 3M \leq 6N$ 이므로 조건에 맞는 그래프를 완성할 수 있습니다.
- 각각의 정점에 연결된 간선의 개수는 $2, 3, 5$ 중 하나이므로 연결된 간선의 개수가 소수여야 한다는 조건도 만족하게 됩니다.
- 시간복잡도는 $\mathcal{O}(3N)$ 입니다.

F. 최솟값을 만들어요

`constructive`, `greedy`
출제진 의도 – **Hard**

- 출제자: `hamuim`

F. 최솟값을 만들어요

- $N = 2^M$ 이라 생각해 봅시다.
- 서로 다른 두 수에 대해 $\oplus$ 연산을 하게되면 항상 0 이 아닌 양의 정수가 됩니다.
- 비용을 가장 적게 사용해 $a_1, a_2, \cdots, a_N$ 을 만들려면 연속적인 두 수는 단 하나의 비트만 다르게 하여 해당 비트에 대한 비용만 사용하게 해야 합니다.
- 처음에 가장 상위의 비트가 같은 것끼리 연속적으로 나열하고 다음으로 같은 것들 안에서 바로 밑의 비트가 같은 것끼리 연속적으로 나열하는 걸 계속 반복하는 게 가장 비용이 적게 들게 됩니다.

F. 최솟값을 만들어요

✓ $N = 2^3$ 인 경우 아래 그림과 같습니다.

	3	2	0	1	5	4	6	7
2nd	0	0	0	0	1	1	1	1
1st	1	1	0	0	0	0	1	1
0th	1	0	0	1	1	0	0	1

F. 최솟값을 만들어요

- $2^0 + 2^1 + \cdots + 2^i < 2^{i+1}$ 이므로 더 이상 비트의 위치를 조정할 필요 없이 현재 상태가 비용을 가장 적게 사용하는 걸 알 수 있습니다.
- 비트별로 관찰해보면 i 번째 비트에서는 0과 1이 2^i 번 나오는 형태가 계속 뒤집히며 나오는게 연속적인 두 수가 하나의 비트만 다르며 비용이 최소가 되는 방법인 걸 찾을 수 있습니다.

F. 최솟값을 만들어요

- $N = 2^M$ 을 토대로 $N \neq 2^M$ 일 때를 생각해 봅시다.
- N 미만의 정수들을 비트별로 0과 1이 몇 개 인지 계산해 이를 $b_{i,0}, b_{i,1}$ 이라 하겠습니다.
- 이들을 순서대로 나열하면서 $a_1, a_2, \cdots, a_N$ 의 i 번째 비트를 결정해 봅시다.

F. 최솟값을 만들어요

- ✓ 수열의 첫 $b_{i,1} \mod 2^i$ 개의 원소의 i 번째 비트는 1 로 다음 $b_{i,0} - (b_{i,1} - b_{i,1} \mod 2^i)$ 개의 원소의 i 번째 비트는 0 으로 설정합니다.
- ✓ 이후로는 0 과 1 을 각각 2^i 개씩 연속으로 나열해 주는데 이전의 관찰을 바탕으로 최대한 0 과 1 이 적게 만나도록 나열하면 됩니다.
- ✓ $b_{i,1} \mod 2^i + b_{i,0} - (b_{i,1} - b_{i,1} \mod 2^i) \geq b_{i-1,1} \mod 2^{i-1} + b_{i-1,0} - (b_{i-1,1} - b_{i-1,1} \mod 2^{i-1})$ 이므로 완성된 수열에는 자연스럽게 0 부터 $N-1$ 까지의 정수들이 나타나게 됩니다.

F. 최솟값을 만들어요

- ✓ 수열의 각 원소에 설정되는 비트는 처음 몇 개를 먼저 나열하고 1과 0을 2^i 개씩 0과 1이 최소로 만나게 설정했기 때문에 $a_1 \oplus a_N$은 항상 일정하게 됩니다.
- ✓ 그러므로 $\sum_{i=1}^{N-1} a_i \oplus a_{i+1}$이 최소가 되게 수열을 만들면 $a_N \oplus a_1$의 값은 수열과 관계없이 항상 일정하기 때문에 방금 만든 수열이 곧 정답이 됩니다.
- ✓ 시간복잡도는 $\mathcal{O}\left(Nlog(N)\right)$입니다.

G. 직각삼각형을 만들어요

`greedy`, `stack`
출제진 의도 – **Hard**

- 출제자: `hamuim`

G. 직각삼각형을 만들어요

- 편의상 텐트를 직각삼각형이라 부르고 천을 왼쪽이나 오른쪽으로 이은 것을 직각삼각형이 왼쪽이나 오른쪽을 가리킨다고 하겠습니다.
- 모든 천을 왼쪽으로 이어 모든 직각삼각형이 왼쪽을 가리키게 했을 때 다른 직각삼각형과 접점이 생기는 것들만 계속 오른쪽으로 방향을 바꿔 접점이 없게 할 수 있다면 정답은 YES입니다.
- 반대로 어떻게든 오른쪽으로 방향을 바꿔도 접점이 존재한다면 정답은 NO입니다.
- 직각삼각형의 빗변의 기울기가 모두 같으므로 한 번 방향이 오른쪽으로 바뀐 직각삼각형은 다시 왼쪽으로 방향을 바꿔도 항상 다른 직각삼각형과 만나버리기 때문에 방향을 되돌릴 필요가 없습니다.

G. 직각삼각형을 만들어요

- 모든 직각삼각형을 접점이 없게 만든다면 이는 어떤 직각삼각형 내부에 있는 직각삼각형들을 하위 노드로 생각하는 트리로 볼 수 있습니다.
- 즉 부모 노드가 같은 직각삼각형들끼리 비교해도 충분합니다.
- 스택을 사용할건데 스택의 원소는 다음과 같습니다.
- { 현재 직각삼각형의 부모 직각삼각형이 오른쪽 방향일 때 높이의 위치(P_x)와 길이(P_y), 부모 직각삼각형 내부에서 더 이상 수직선 위에 위치할 수 없는 구간들을 원소로 가지는 스택(P_{stack}) }

G. 직각삼각형을 만들어요

- 처음에 모든 직각삼각형을 내부에 둘 수 있는 오른쪽 방향의 매우 큰 직각삼각형이 있다고 생각하고 이를 스택에 넣습니다.
- 직각삼각형의 높이의 위치가 작은 순으로 보면서 방향을 정해줍시다.
- 현재 높이의 위치가 작은순으로 i 번째 직각삼각형의 차례이고 i 번째 직각삼각형의 높이의 위치가 i_x 이고 길이가 i_y 이며 스택의 가장 위에 있는 원소가 i 번째 직각삼각형의 부모로 생각하면 됩니다.

G. 직각삼각형을 만들어요

- 부모 직각삼각형의 방향은 오른쪽이므로 스택의 가장 위에 있는 직각삼각형의 오른쪽 끝점이 i_x 이상이 될 때까지 스택에서 빼주어 스택의 가장 위에 있는 원소가 i 번째 직각삼각형의 부모가 되게 해줍니다.
- 부모 직각삼각형에 대해 $i_x \leq P_x + M \cdot P_y \leq i_x + M \cdot i_y$ 라면 i 번째 직각삼각형의 방향과 상관없이 항상 부모 직각삼각형의 빗변과 i 번째 직각삼각형의 높이 사이에서 접점이 생기기 때문에 답은 NO입니다.
- 또한 P_{stack} 에서 i_x 가 구간에 포함되는지 확인해 구간에 포함되어도 부모 직각삼각형 내부에서 이전의 직각삼각형과 접점이 생기기 때문에 답은 NO입니다.

G. 직각삼각형을 만들어요

- i번째 직각삼각형의 방향을 확인하기 전에 하나 생각해 봅시다.
- i번째 직각삼각형의 방향이 왼쪽이라면 이후의 직각삼각형들은 높이가 i_x보다 뒤에 위치하기 때문에 $[i_x - M \cdot i_y, i_x]$ 구간에 빗변을 내릴 수 없습니다.
- i번째 직각삼각형의 방향이 오른쪽이라면 이후의 직각삼각형들은 $[i_x - M \cdot i_y, i_x]$ 구간뿐만 아니라 $[i_x, i_x + M \cdot i_y]$ 구간에도 빗변을 내릴 수 없습니다.

G. 직각삼각형을 만들어요

- 이제 i 번째 직각삼각형이 방향을 왼쪽으로 할 수 있는지 확인합니다.
- 이는 P_{stack} 에서 $[i_x - M \cdot i_y, i_x]$ 구간이 완전히 포함하는 구간들을 전부 빼준 다음 가장 위에 있는 원소가 $i_x - M \cdot i_y$ 를 포함하는지 확인하면 됩니다.
- 포함하지 않는다면 i 번째 직각삼각형의 방향을 왼쪽으로 할 수 있으므로 P_{stack} 의 가장 위에 $[i_x - M \cdot i_y, i_x]$ 를 넣어주고 i 번째 직각삼각형의 방향을 L 로 설정합니다.

G. 직각삼각형을 만들어요

- 방향을 왼쪽으로 할 수 없다면 오른쪽 방향이므로 마찬가지로 P_{stack} 의 가장 위의 원소와 $[i_x - M \cdot i_y, i_x + M \cdot i_y]$ 구간을 합쳐주고 i 번째 직각삼각형의 방향을 R 로 설정합니다.
- 다음으로 i 번째 직각삼각형의 방향이 오른쪽이므로 i 번째 직각삼각형이 부모인 경우를 확인하기 위해 i 번째 직각삼각형을 스택의 가장 위에 원소로 넣어줍니다.
- i 번째 직각삼각형의 방향을 왼쪽으로 했을 경우 이후의 직각삼각형들의 높이의 위치가 i_x 보다 크기 때문에 더 이상 직각삼각형이 내부에 존재할 수 없어 스택에 넣어주지 않아도 됩니다.

G. 직각삼각형을 만들어요

- 이를 마지막 직각삼각형까지 반복해 모든 직각삼각형의 방향을 구할 수 있습니다.
- 모든 과정에서 P_{stack} 에 구간을 최대 N 번 삽입하고 오른쪽 방향의 직각삼각형을 스택에 최대 N 번 삽입하게 됩니다.
- 시간복잡도는 $\mathcal{O}(N)$ 입니다.
- 별해로 세그먼트 트리를 사용해 직각삼각형의 빗변을 내릴 구간을 계속 업데이트하며 방향을 정하는 풀이도 있습니다.

H. 나머지를 만들어요

`ad_hoc`, `binary_search`, `case_work`, `constructive`
출제진 의도 – **Challenging**

- 출제자: `hamuim`

H. 나머지를 만들어요

- $[s, e]$를 N이 있을 수 있는 구간이라고 정의하고 처음에 $s = 0, e = 10^9$로 설정하겠습니다.
- 임의의 정수 X에 대한 쿼리의 답변을 $Q(X)$라 하겠습니다.
- $s = 0, e > 2$인 경우에서 $mid = \left\lfloor \frac{s + e + 1}{2} \right\rfloor$라 하고 $Q(e) = Q(e - 1)$를 만족하면 N이 s에 더 가깝다는 뜻이기 때문에 $e = mid$가 되고 아니면 $s = mid$가 됩니다.
- $M = 1$인 경우에는 항상 $Q(X) = 0$이 되어 N의 값이 상관없어지므로 $N = 0$으로 생각하면 위 조건을 만족하게 됩니다.
- 우선은 이 방법으로 $s > 0$이 될 때까지 이분 탐색을 해봅시다.

H. 나머지를 만들어요

- $s = 0$일 때 방금의 연산을 계속하여 $e = 2$가 된다면 $Q(2)$와 $Q(3)$을 통해 N을 알 수 있습니다.
- $Q(2) = 0, Q(3) = 0$이라면 $N = 0$이고 아니라면 $[s, e]$를 $[1, 2]$로 설정해주면 됩니다.
- 앞서 말했듯이 $N = 0$이라면 M과 관계없이 쿼리의 답변이 항상 0이기 때문에 M의 최솟값은 1이고 정답은 $0, 1$이 되며 $M = 1$이어도 마찬가지입니다.

H. 나머지를 만들어요

- 이후의 단계로 진행하기 전에 고려해야 하는 부분이 몇 가지 존재합니다.
- 먼저 $e = s + 1$이 되었다면 $Q(e) = 0$이라면 $N = e$이고 아니라면 $N = s$가 되어 N을 찾을 수 있습니다.
- 다음으로 $mid = \left\lfloor \frac{s + e + 1}{2} \right\rfloor$로 설정하였기 때문에 $s > 0$이라면 $e - s \leq s$가 성립합니다.
- 이를 계속 활용해 봅시다.

H. 나머지를 만들어요

- 아까의 $s = mid$로 s가 처음으로 0이 아니게 된다면 먼저 $M = 2$인지를 확인하여 따로 처리해 주어야 합니다.
- 단 $s = 1$인 경우는 방금 $e = s + 1$이 되기에 제외해도 되며 $M = 2$인 경우를 따로 처리하는 이유는 나중에 설명하겠습니다.
- $mid = \left\lfloor \frac{s + e + 1}{2} \right\rfloor$라 하면 $mid, mid + 1, mid + 2, mid + 3$를 통해 $M = 2$인지를 판단할 수 있습니다.
- $\sum_{i=0}^{3} Q(mid + i) \leq 2$라면 $M = 2$가 되며 증명은 다음과 같습니다.

H. 나머지를 만들어요

- ✓ $s = 2, e = 4$인 경우 가능한 모든 N에 대해 확인하면 다음과 같습니다.
- ✓ $N = 2, M = 2$인 경우 $Q(mid) = 1, Q(mid + 1) = 0, Q(mid + 2) = 0, Q(mid + 3) = 0$입니다.
- ✓ $N = 2, M > 2$인 경우 $Q(mid) = 1, Q(mid + 1) = 2$입니다.
- ✓ $N = 3, M = 2$인 경우 $Q(mid) = 0, Q(mid + 1) = 1, Q(mid + 2) = 0, Q(mid + 3) = 1$입니다.

H. 나머지를 만들어요

- $N = 3, M > 2$인 경우 $Q(mid + 1) = 1, Q(mid + 2) = 2$입니다.
- $N = 4, M = 2$인 경우 $Q(mid) = 1, Q(mid + 1) = 0, Q(mid + 2) = 1, Q(mid + 3) = 0$입니다.
- $N = 4, M > 2$인 경우 $Q(mid + 2) = 1, Q(mid + 3) = 2$입니다.

H. 나머지를 만들어요

- 다음으로 $s = 4, e = 8$에서 $N = 4$인 경우를 보겠습니다.
- $M = 2$인 경우 $Q(mid) = 0, Q(mid + 1) = 1, Q(mid + 2) = 0, Q(mid + 3) = 0$입니다.
- $M = 3$인 경우 $Q(mid) = 2, Q(mid + 2) = 1$입니다.
- $M > 3$인 경우 $Q(mid) = 2, Q(mid + 1) = 3$입니다.

H. 나머지를 만들어요

- 남은 모든 경우에서 $e - s \leq s$ 이므로 $mid, mid+1, mid+2, mid+3$ 이 N 과의 차이가 가장 작아지므로 이때 $Q(X) = |N - X| \mod M$ 로 생각할 수 있습니다.
- $(Q(mid), Q(mid+1), Q(mid+2), Q(mid+3))$ 로 가능한 형태는 $M = 2$ 인 경우 $(0,1,0,1), (1,0,1,0)$ 입니다.
- $M > 2$ 인 경우 임의의 양의 정수 k 에 대해 $(k, k+1, k+2, k+3), (k, k+1, k+2, 0), (k, k+1, 0, 1), (k, 0, 1, 2), (0, 1, 2, 3), (1, 0, 1, 2)$ 혹은 이를 반대로 뒤집은 형태입니다.
- 그러므로 항상 $M = 2$ 일 때만 $\sum_{i=0}^{3} Q(mid+i) \leq 2$ 가 성립하는 것을 알 수 있습니다.

H. 나머지를 만들어요

- $M = 2$인 경우는 나중으로 미루고 $M > 2$인 경우를 먼저 보겠습니다.
- 이제 $s > 1$이고 $mid = \left\lfloor \frac{s+e+1}{2} \right\rfloor$라 하면 N은 항상 $Q(mid), Q(mid+1)$에서 $mid, mid+1$과의 차이가 최솟값이 됩니다.
- $Q(mid) > 0$이라면 $Q(mid+1)$를 확인해 $Q(mid+1)$이 $Q(mid)$보다 줄어드는지로 s와 e 중에서 어디와 가까운지 알 수 있습니다.
- $Q(mid) = 0$이라면 $Q(mid+1)$이 1인지로 s와 e중에서 어디와 가까운지 알 수 있고 이를 $e \leq s+1$가 될 때까지 반복해 범위를 줄여 N을 찾을 수 있습니다.
- $M = 2$라면 쿼리의 결과가 항상 0과 1이라 N의 위치를 판단할 수 없기 때문에 미리 예외로 제외했습니다.

H. 나머지를 만들어요

- ✓ N을 찾은 상태에서 가능한 M의 최솟값을 찾아야 하는데 그전에 쿼리로 질문했을 때 $\bmod M$을 하기 전 최댓값을 생각해 봅시다.
- ✓ 대부분 $Q(2N)$에서 N과의 차이가 N이 되게 할 때 가장 크지만 $2N > 10^9$가 되버리면 $Q(10^9)$나 X와 $2X$의 중점이 N이 되도록 하는 $Q(\left\lfloor \frac{2N+2}{3} \right\rfloor - 1)$에서의 차이가 더 클 수도 있습니다.

H. 나머지를 만들어요

- ✓ $N \le \frac{3}{4}10^9$ 라면 쿼리에서 $\bmod M$ 을 하기 전 최댓값이 항상 $10^9 - N$ 보다 작기 때문에 $N+2 \le X \le 10^9+1$ 범위에서 $Q(X)$ 가 0 이며 N 과의 차이가 가장 작은 X 에 대해 $X-N$ 이 M 의 최솟값이 됩니다.
- ✓ 이는 $s = N+1, e = \min(2N+1, 10^9+1)$ 로 설정하고 이분 탐색을 하여 찾을 수 있습니다.

H. 나머지를 만들어요

- ✓ $N > \frac{3}{4}10^9$ 라면 $10^9 - N$ 보다 $N - \frac{1}{3}N$ 의 값이 더 커지므로 $\left\lfloor \frac{2N+2}{3} \right\rfloor - 1 \leq X \leq N - 2$ 범위에서 $Q(X)$ 가 0 이며 N 과의 차이가 가장 작은 X 에 대해 $N - X$ 이 M 의 최솟값이 됩니다.
- ✓ 이는 $s = \left\lfloor \frac{2N+2}{3} \right\rfloor - 1, e = N - 1$ 로 설정하고 이분 탐색을 하여 찾을 수 있습니다.

H. 나머지를 만들어요

- 아까 마무리하지 못한 $M = 2$인 경우를 다시 보겠습니다.
- 이때 쿼리의 답변은 항상 0이나 1이며 추가적인 쿼리 없이 이전에 확인한 $Q(mid), Q(mid + 1)$의 홀짝성으로 N이 홀수인지 짝수인지를 알 수 있습니다.

H. 나머지를 만들어요

- N의 홀짝을 판단하였다면 s, e도 ± 1을 더하여 홀짝을 조절해 주면 됩니다.
- 쿼리로 질문할 때 $Q(X)$에서 X가 짝수라면 $iX \leq N \leq (i+1)X$에서 가장 가까운 값과의 차이에 $\mod 2$를 하면 똑같기 때문에 위치를 판단할 수 없습니다.
- X가 홀수라면 쿼리의 홀짝성으로 N이 s와 e 중에서 어느 쪽과 가까운지 알 수 있으므로 쿼리에서 X는 홀수로 질문해야 합니다.

H. 나머지를 만들어요

- ✓ $e \leq 5 \cdot 10^8$ 인 경우에는 $0 \leq s < e \leq X$ 가 되게 질문해서 이분 탐색으로 s, e 의 구간을 계속 절반으로 좁혀나갈 수 있습니다.
- ✓ N 의 홀짝을 알기 때문에 항상 s 와 e 의 홀짝이 같고 $e \geq s + 2$ 가 되게 구간을 유지할 수 있으므로 구간을 $s = e$ 가 될 때까지 반으로 줄여 N 을 찾을 수 있습니다.

H. 나머지를 만들어요

- $e > 5 \cdot 10^8$ 인 경우에는 $X \leq s < e \leq 2X$ 가 되게 질문해서 이분 탐색으로 s, e 의 구간을 계속 절반으로 좁혀 나갈 수 있습니다.
- 마찬가지로 N 의 홀짝을 알기 때문에 항상 s 와 e 의 홀짝이 같고 $e \geq s + 2$ 가 되게 구간을 유지할 수 있습니다.
- 하지만 $s + 2 = e$ 일 때 $s + 1$ 이 3 의 배수라면 $X \leq s < e \leq 2X$ 에서 $Q(X)$ 로 질문해도 s 와 e 가 둘 다 X 와 가깝거나 $2X$ 와 가까운 상황이 생길 수도 있습니다.

H. 나머지를 만들어요

- 이를 해결하기 위해서는 홀수이고 $iX < s < e < (i+1)X$ 에서 $s - iX < (i+1)X - s$ 와 $(i+1)X - e < e - iX$ 를 만족하는 X 를 찾아야 합니다.
- 하지만 이는 너무 어렵기에 다른 방법인 $\frac{(2i+1)X}{2}$ 가 $s + \frac{1}{2}$ 이나 $s + \frac{3}{2}$ 이 되게 하는 X 를 찾아봅시다.
- 2 를 곱해주면 $2s+1$ 이나 $2s+3$ 은 홀수이므로 소수가 아니라면 홀수 2 개로 나눠 $2i+1$ 과 X 에 해당하는 수를 얻어 s 와 $s+2$ 중 N 을 찾을 수 있습니다.
- 그렇지 않다면 s 와 $s+2$ 중에서 쿼리로 N 을 판별할 수 없기 때문에 N 의 최솟값은 s 가 됩니다.

H. 나머지를 만들어요

- ✓ $M = 2$로 미리 제외한 경우가 아니라면 N을 찾기까지 $2log(10^9)$개의 쿼리가 필요하고 M을 찾기까지 $log(10^9)$개의 쿼리가 필요합니다.
- ✓ $M = 2$로 미리 제외한 경우에서는 $M = 2$를 판별하기까지 최대 $2log(10^9)$개에 가까운 쿼리가 필요하고 N을 찾기까지 $log(10^9)$개의 쿼리가 필요합니다.
- ✓ 약 $3log(10^9)$개의 쿼리를 사용해 문제를 해결할 수 있습니다.