1. 2인승 조정 (4점)

2인승 조정 경기에서 한 팀은 왼쪽 자리를 맡을 선수 한 명과 오른쪽 자리를 맡을 선수 한 명으로 이루어진다.

왼쪽 자리에 배치할 수 있는 후보 8명의 능력치는 각각 $45, 40, 45, 41, 46, 41, 41, 40$이고, 오른쪽 자리에 배치할 수 있는 후보 8명의 능력치는 각각 $14, 13, 24, 17, 17, 16, 14, 13$이다.

왼쪽 후보 한 명과 오른쪽 후보 한 명을 골라 팀을 구성하고, 선발된 두 후보의 능력치 차(왼쪽 선수의 능력치 오른쪽 선수의 능력치)를 D라 하자. $D \leq 20$인 모든 팀 구성 가운데, D의 최댓값은?

정답: 17

2. 신발 찾기 (4점)

신발 한 켤레는 왼쪽과 오른쪽 두 짝으로 이루어진다. (즉, '짝'은 신발 한 쪽을 뜻한다.) 25켤레의 신발, 즉 50짝이 있으며, 50짝은 모두 서로 구별된다.

이 50짝 가운데 서로 다른 세 짝을 임의로 고를 때, 고른 세 짝 가운데 같은 켤레에 속한 두 짝이 포함될 확률은?

- ≫ $\frac{1}{98}$
- ≫ $\frac{1}{49}$
- ≫ $\frac{2}{49}$
- ≫ $\frac{3}{49}$ (정답)
- ≫ $\frac{4}{49}$
- ≫ $\frac{5}{49}$
- ≫ $\frac{6}{49}$
- ≫ $\frac{1}{50}$
- ≫ $\frac{1}{25}$
- ≫ $\frac{3}{50}$

3. 1과 2 (4점)

아래 조건을 모두 만족하는 양의 정수 N 가운데 가장 작은 값을 구하라.

- N의 십진수 표현은 숫자 1과 2로만 이루어져 있다.
- N의 십진수 표현에서 숫자 1이 정확히 8번 등장한다.
- N의 십진수 표현에서 숫자 2가 정확히 8번 등장한다.
- N의 십진수 표현에서 연속한 부분 문자열 '12'가 정확히 4번 등장한다.
- N의 십진수 표현에서 연속한 부분 문자열 '21'이 정확히 5번 등장한다.

정답: 2111121212122221

4. 비트 방정식 (6점)

0 이상의 정수 a, b에 대하여 $a \oplus b$는 a와 b의 **비트별 배타적 논리합(XOR)** 이다. 즉, a와 b의 이진수 표현에서 같은 자리의 두 비트가 서로 다르면 해당 자리에 1을, 같으면 0을 두어 얻는 값이다. (C, C++, Java, Python 언어의 ^ 연산자와 같다.)

예를 들어,

- $6 \oplus 3$의 경우, $6 = 110_{(2)}$, $3 = 011_{(2)}$이므로 $6 \oplus 3 = 101_{(2)} = 5$이다.
- $10 \oplus 12$의 경우, $10 = 1010_{(2)}$, $12 = 1100_{(2)}$이므로 $10 \oplus 12 = 0110_{(2)} = 6$이다.

양의 정수 n과 k에 대하여, $n \star k$는 n을 k로 나눈 몫을 뜻한다. 예를 들어, $27 \star 4 = 6$이다.

다음 식을 만족하는 양의 정수 n이 유일하게 존재한다. 우변의 값이 이진법으로 표현되었음에 유의하라.

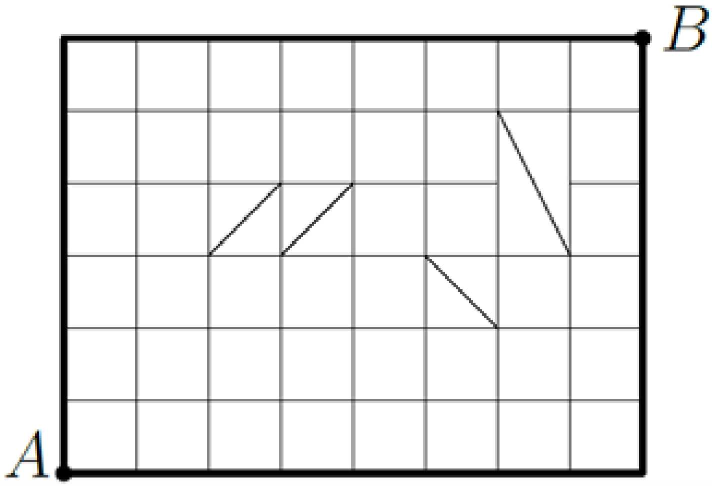
$$n \oplus (n \star 2^2) \oplus (n \star 2^5) = 10011110_{(2)}$$

이 n을 **이진법으로** 나타내어라.

정답: 10110110

5. 가장 짧은 경로 (7점)

아래 그림은 A 지점과 B 지점 주변을 나타낸 약도이다. 그림에 있는 모든 선분은 사람이 지나갈 수 있는 길을 뜻한다. 그림 속 단위 정사각형 한 칸의 한 변의 길이는 1이다.



A 지점에서 B 지점까지 가는 최단 경로의 가짓수를 구하라.

정답: 420

6. 스트라이크 (8점)

민수와 그의 친구 지수가 다음과 같은 게임을 한다. 민수가 5자리 자연수 하나를 정하면, 지수는 그 수를 맞히기 위해 여러 번 추측한다. 민수는 지수가 추측할 때마다, 숫자가 일치하는 자리의 개수를 알려 준다.

예를 들어, 민수가 정한 수가 12345이고 지수가 10394를 추측했다면, 첫 번째 자리의 1과 세 번째 자리의 3이 일치하므로 알려 주는 개수는 2이다.

지수의 추측과 그 결과가 아래 표와 같을 때, 민수가 정한 5자리 수를 구하라. 정답은 유일하게 결정된다.

지수의 추측	일치하는 자리의 개수
40758	0
61409	1
11354	2
86143	1
52752	2
57531	1
58879	1

정답: 51342

7. 꼬리합 (8점)

십진법으로 나타냈을 때 8자리인 양의 정수 N의 **꼬리합**은 N의 왼쪽 끝자리부터 0개, 1개, ..., 7개의 숫자를 지워 얻는 8개의 수의 합이다. 숫자를 지운 결과의 앞자리에 생기는 0은 무시하고 수로 본다.

예를 들어, 40891234의 꼬리합은 $40891234 + 0891234 + 891234 + 91234 + 1234 + 234 + 34 + 4 = 42766442$이다.

꼬리합이 20260510이면서 십진법으로 나타냈을 때 8자리인 양의 정수 하나를 구하라. 조건을 만족하는 답이 여러 개 존재할 수 있으며, 그중 어느 것을 답해도 정답으로 인정한다.

정답: 14737915, 15063915, 20063915 (이 중 하나)

8. 이진 트리의 거리 (9점)

루트가 있는 이진 트리를 생각하자. 이 트리의 단말 정점은 8개이며, 모든 내부 정점에는 정확히 두 개의 자식이 있다.

단말 정점에는 트리를 중위 순회할 때 방문되는 순서대로 $1, 2, \ldots, 8$의 번호가 붙어 있다.

두 단말 정점 u, v 사이의 **거리** $d(u, v)$는 u에서 v로 가는 단순 경로에 포함된 간선의 수로 정의한다.

이 트리에서 각 $1 \le i \le 7$에 대한 $d(i, i+1)$의 값은 아래 표와 같다.

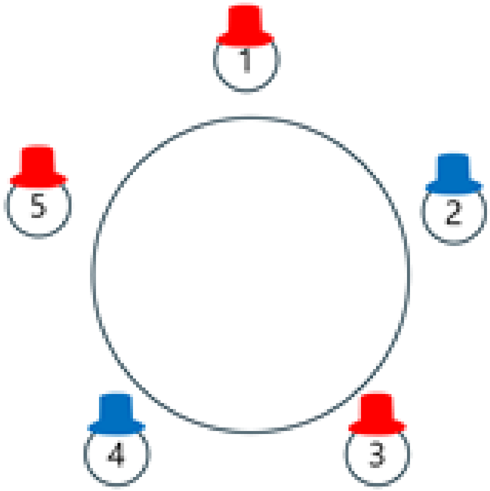
i	1	2	3	4	5	6	7
$d(i, i+1)$	2	6	3	2	4	3	4

위 표의 값을 만족하는 트리는 유일하게 결정된다. $d(1, 5) + d(3, 7) + d(2, 8)$의 값을 구하라.

정답: 15

9. 모자 색 맞추기 (9점)

TV 쇼 게임에 참가한 5명의 사람이 아래 그림과 같이 원탁에 둘러앉아 있다. 편의상 각 사람을 1번부터 5번까지의 번호로 구분하자.



게임은 다음과 같이 진행된다.

- 빨간 모자 4개와 파란 모자 3개, 총 7개의 모자 가운데 사회자가 5개를 골라 참가자들에게 하나씩 씌운다.
- 그림과 같이 1번 참가자는 빨간 모자, 2번 참가자는 파란 모자, 3번 참가자는 빨간 모자, 4번 참가자는 파란 모자, 5번 참가자는 빨간 모자를 쓰고 있다.
- 참가자들은 전체 모자 구성이 빨간 모자 4개, 파란 모자 3개라는 사실과, 그중 5개가 자신들에게 씌워졌다는 사실을 안다. 그러나 어떤 5개가 선택되었는지는 알지 못한다.
- 각 참가자는 자기 모자의 색은 볼 수 없지만, 다른 참가자들의 모자 색은 볼 수 있다.
- 사회자는 1번 참가자부터 시작하여 번호 순서대로 "당신은 자신의 모자 색을 알 수 있습니까?"라고 묻는다.
- 질문을 받은 참가자는 자신의 모자 색을 정확히 알 수 있으면 'Yes', 그렇지 않으면 'No'라고 답한다.
- 모든 참가자는 이전 참가자들의 답을 모두 들으며, 완벽하게 논리적으로 추론한다.

1번 참가자는 'No'라고 답했다.

이때 2번, 3번, 4번 참가자의 대답을 순서대로 나열한 것은?

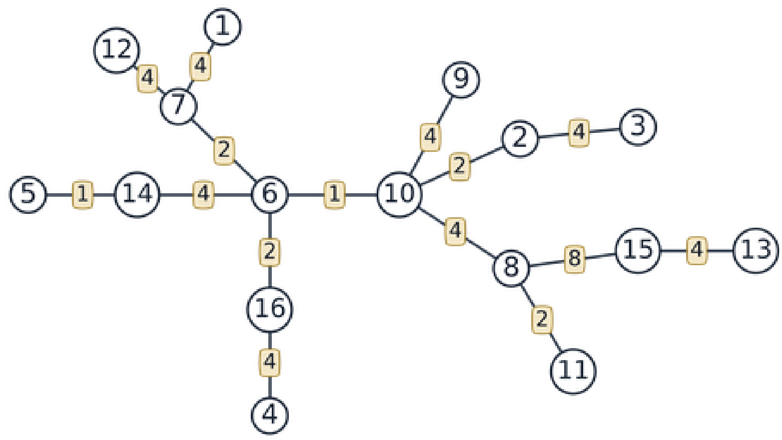
- ≫ No, No, No
- ≫ No, No, Yes
- ≫ No, Yes, No
- ≫ No, Yes, Yes (정답)
- ≫ Yes, No, No

- ≫ Yes, No, Yes
- ≫ Yes, Yes, No
- ≫ Yes, Yes, Yes

10. 도시 순회 (12점)

1번부터 16번까지 번호가 붙은 16개의 도시가 있다. 서로 다른 두 도시 사이에는 도로가 하나씩 있어 완전 그래프를 이룬다.

그중 아래 그림에 표시된 15개의 도로는 하나의 트리를 이루며, 각 도로의 이동 시간은 그림에 적힌 값이다.



그림에 표시되지 않은 나머지 도로의 이동 시간은 모두 6이다.

모든 도시를 정확히 한 번씩 방문하는 경로의 총 이동 시간의 최솟값을 구하라. 경로의 시작 도시와 끝 도시는 자유롭게 고를 수 있다.

정답: 58

11. 만나는 반직선 (12점)

좌표평면 위에서 x좌표가 1 이상 4 이하, y좌표가 1 이상 5 이하인 정수 좌표의 점 20개를 생각하자. 동, 서, 남, 북 방향은 각각 $+x$, $-x$, $-y$, $+y$ 방향을 뜻한다.

두 사람 A와 B는 각각 독립적으로 20개의 점 가운데 하나와 동, 서, 남, 북 가운데 한 방향을 고른 뒤, 선택한 점에서 선택한 방향으로 반직선을 그린다. A와 B는 같은 점과 같은 방향을 선택할 수도 있다.

한 사람이 선택할 수 있는 경우의 수는 $20 \times 4 = 80$가지이므로, 전체 경우의 수는 $80 \times 80 = 6\,400$가지이다.

이 가운데 두 반직선에 모두 포함되는 점이 존재하는 경우의 수를 구하라.

정답: 1780

12. 각기둥 그래프 (15점)

16각기둥 그래프를 다음과 같이 정의한다. 꼭짓점은 위쪽 16각형의 꼭짓점 16개와 아래쪽 16각형의 꼭짓점 16개, 총 32개이다. 간선은 다음 세 종류이다.

- 위쪽 16각형의 변 16개
- 아래쪽 16각형의 변 16개
- $1 \leq i \leq 16$인 각 i에 대하여, 위쪽 16각형의 i번째 꼭짓점과 아래쪽 16각형의 i번째 꼭짓점을 잇는 세로 선분

따라서 간선은 총 48개이다.

이 그래프의 각 꼭짓점에 0 또는 1을 적는 방법을 **배치**라 하자. 이 32개의 꼭짓점은 서로 구별되므로, 가능한 배치는 모두 2^{32}가지이다.

아래 두 조건을 모두 만족하는 배치의 개수를 구하라.

- 각 꼭짓점에 적힌 값은, 그 꼭짓점과 간선으로 연결된 세 꼭짓점에 적힌 값 가운데 두 번 이상 나타나는 값과 같다.
- 1이 적힌 꼭짓점이 정확히 16개이다.

정답: 414

13. 조 나누기 (8점)

1부터 16까지의 자연수가 있다. 이 수들을 4개의 조(가, 나, 다, 라)에 4개씩 나누어 넣으려고 한다.

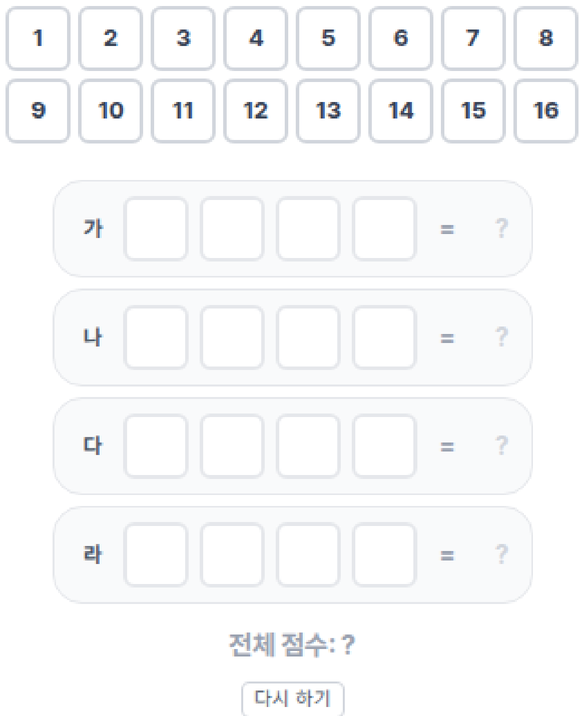
각 조의 **점수**는 그 조에 속한 수 중 가장 작은 수와 가장 큰 수의 합이다. 예를 들어, 한 조에 $\{2, 5, 11, 14\}$가 있다면 그 조의 점수는 $2 + 14 = 16$이다.

전체 점수는 4개 조의 점수를 모두 더한 값이다.

전체 점수를 **최소화**하는 배치를 찾아라.

배치할 수를 클릭하여 선택한 뒤, 원하는 조의 빈 칸을 클릭하면 해당 위치에 배치된다. 배치된 수를 클릭하면 다시 되돌릴 수 있다. 각 조의 점수는 조 오른쪽에 표시되며, 하단에는 전체 점수가 표시된다.

“다시 하기” 버튼을 누르면 배치를 초기화할 수 있다.



각 조의 점수는 그 조에 넣은 수들 중 가장 작은 수와 가장 큰 수의 합이다. 네 조의 점수를 모두 더한 값이 전체 점수이며, 이를 가능한 한 작게 만드는 것이 목표이다.

1부터 16까지를 네 조에 4개씩 모두 배치했을 때 전체 점수가 56이면 만점이다.

라 조에 $4, 5, 6, 7$, 다 조에 $3, 8, 9, 10$, 나 조에 $2, 11, 12, 13$, 가 조에 $1, 14, 15, 16$을 넣으면 각 조의 점수가 11, 13, 15, 17이 되어 합이 56이다. 위 화면은 그런 배치의 한 예이다.

위의 방법 이외에도 전체 점수가 56이 되게 배치하면 정답 처리된다.

14. 직사각형 만들기 (9점)

12개의 막대기가 있고, 각 막대기가 아래에 있는 수평선에 놓여 있다. 각 막대기의 길이는 막대기에 적혀 있다.

이 막대기들을 두 묶음으로 나누어 직사각형을 만들려고 한다. 왼쪽 묶음에 속한 막대기들의 길이의 합이 직사각형의 세로가 되고, 오른쪽 묶음에 속한 막대기들의 길이의 합이 가로가 된다.

막대기들을 적절히 나누어, 만들어지는 직사각형의 넓이를 최대화하라. 빈 묶음의 경우 길이의 합은 0으로 간주한다.

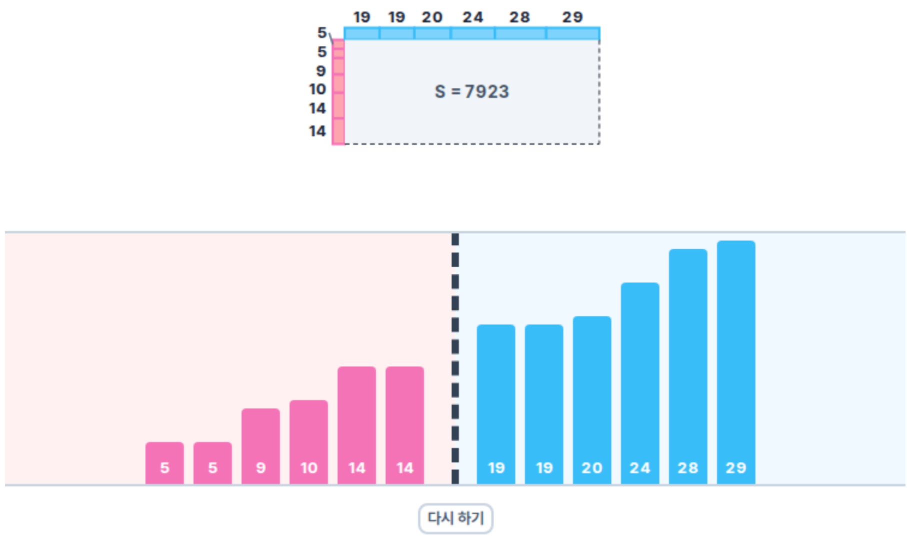
조작 방법

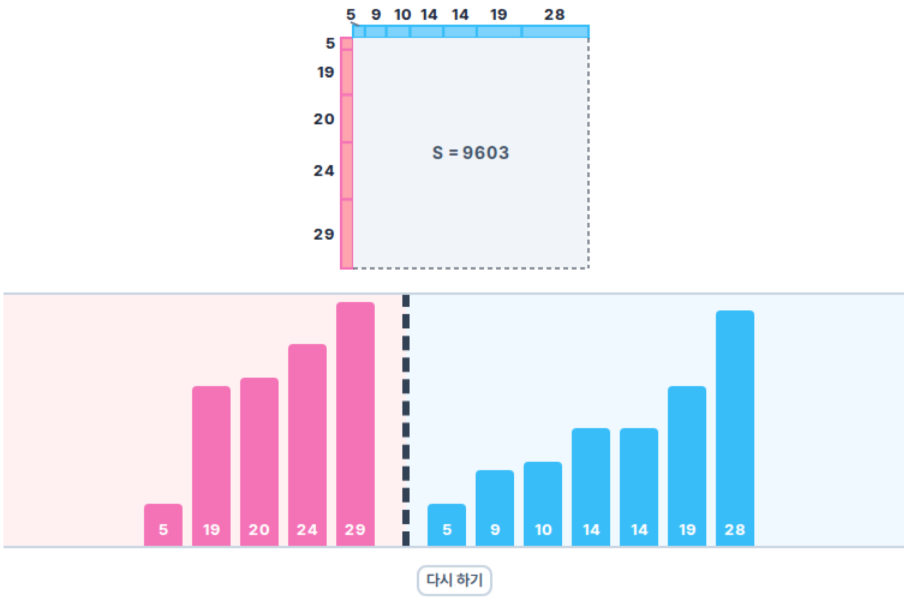
막대기들이 구분선을 기준으로 왼쪽 묶음(분홍색)과 오른쪽 묶음(하늘색)으로 나뉘어 표시된다. 각 막대기의 높이는 길이에 비례하며, 길이가 수로 표기되어 있다. 현재 묶음에 따라 만들어지는 직사각형의 모양이 상단에 표시된다.

막대기를 드래그하여 원하는 묶음으로 옮길 수 있다. 구분선을 넘겨 다른 묶음으로 이동시킬 수도 있다. “다시 하기” 버튼을 눌러 처음 상태로 되돌릴 수 있다.

채점 기준

직사각형의 넓이가 최댓값이면 전체 점수의 100%를 얻는다.





막대기 길이의 총합은 196이다. 한쪽 묶음의 길이 합을 a라 하면 다른 쪽 묶음의 길이 합은 $196-a$이고, 넓이는 $a(196-a)$이다. 이 값은 a가 98에 가까울수록 커진다.

가능한 묶음의 길이 합 중 98에 가장 가까운 값은 97과 99이다. 예를 들어 왼쪽 묶음에 길이 $5, 19, 20, 24, 29$인 막대기를 놓으면 합이 97이 되고, 오른쪽 묶음에는 나머지 $5, 9, 10, 14, 14, 19, 28$인 막대기가 남아 합이 99가 된다. 이때 넓이는

$$97 \times 99 = 9603$$

이다.

위의 방법 이외에도 두 묶음의 합이 각각 97과 99가 되도록 나누면 정답 처리된다.

15. 양팔 저울 (10점)

양팔 저울과 10개의 추가 있다. 추의 무게는 각각 $7, 10, 11, 12, 14, 16, 18, 20, 21, 22$이다.

추를 하나씩 골라 왼쪽 또는 오른쪽 접시에 올리려고 한다. 아래와 같이 $<$와 $>$로만 이루어진 길이 10의 문자열이 주어진다.

$$< > < > < > < > > <$$

i번째로 추를 올린 직후, 저울의 기울기가 문자열의 i번째 문자와 일치해야 한다.

- $>$: 왼쪽 접시가 더 무거움 (왼쪽으로 기울어짐)
- $<$: 오른쪽 접시가 더 무거움 (오른쪽으로 기울어짐)

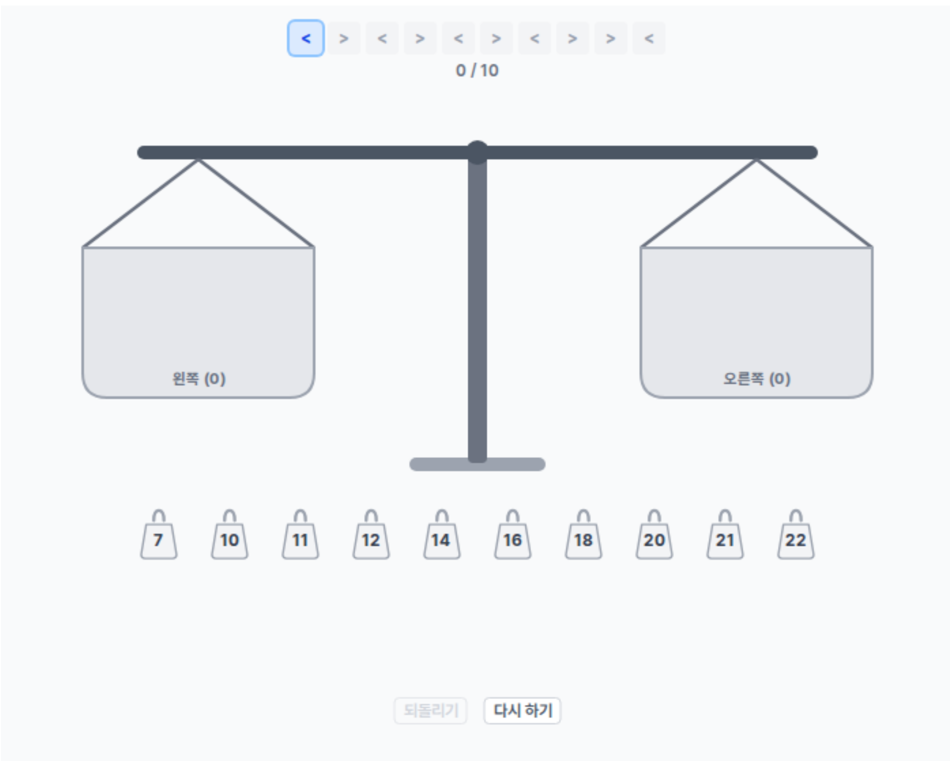
예를 들어, 첫 번째로 무게 7인 추를 오른쪽 접시에 올리면 오른쪽이 더 무거우므로 기울기는 $<$이다. 문자열의 첫 번째 문자가 $<$이므로 조건에 맞는다. 이어서 두 번째로 무게 22인 추를 왼쪽 접시에 올리면 왼쪽 총무게는 22, 오른쪽 총무게는 7이므로 기울기는 $>$이다. 문자열의 두 번째 문자가 $>$이므로 역시 조건에 맞는다.

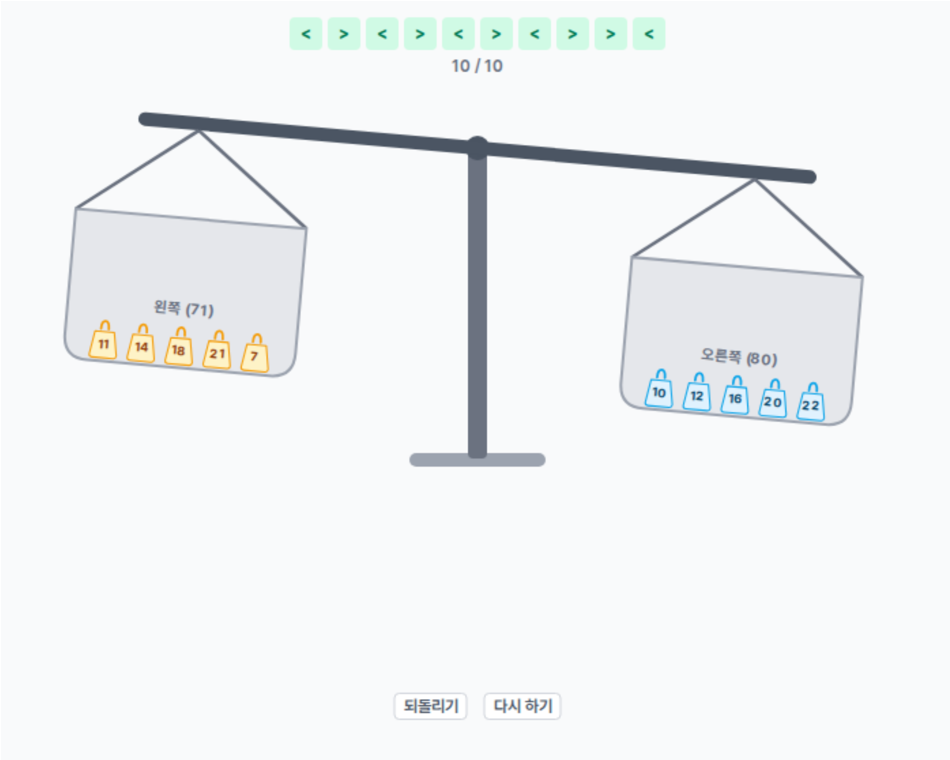
이와 같이 10개의 추를 모두 올려서 매 단계마다 조건을 만족하도록 하라.

아직 올리지 않은 추 중 하나를 클릭하여 선택한 뒤, 왼쪽 또는 오른쪽 접시를 클릭하여 배치할 수 있다. “되돌리기” 버튼을 누르면 마지막 배치를 되돌릴 수 있고, “다시 하기” 버튼을 누르면 처음 상태로 되돌릴 수 있다.

채점 기준

10개의 추를 모두 올렸고, 매 단계마다 저울의 기울기가 주어진 문자열과 일치하면 전체 점수의 100%를 받는다.





다음 순서대로 추를 올리면 된다.

1. 10을 오른쪽에 올린다.
2. 11을 왼쪽에 올린다.
3. 12를 오른쪽에 올린다.
4. 14를 왼쪽에 올린다.
5. 16을 오른쪽에 올린다.
6. 18을 왼쪽에 올린다.
7. 20을 오른쪽에 올린다.
8. 21을 왼쪽에 올린다.
9. 7을 왼쪽에 올린다.
10. 22를 오른쪽에 올린다.

각 단계가 끝난 뒤 저울의 기울기는 차례대로

$$< > < > < > < > > <$$

와 같다. 따라서 지문에 주어진 <><><><>><와 모두 일치한다.

위의 방법 이외에도 10개의 추를 모두 올리는 동안 매 단계의 기울기가 주어진 문자열과 일치하면 정답 처리된다.

16. 뒤집기 (15점)

8개의 숫자 카드가 일렬로 놓여 있다. 처음에 카드는 $[8,7,6,5,4,3,2,1]$의 순서로 놓여 있다.

한 번의 조작으로 카드 하나를 골라 정확히 **2칸** 앞 또는 뒤로 이동시킬 수 있다. 카드를 이동하면, 사이에 있는 카드들은 빈 자리를 채우도록 한 칸씩 밀려난다.

예를 들어 $[8,7,6,5,4,3,2,1]$에서 맨 앞의 8을 2칸 뒤로 이동하면 $[7,6,8,5,4,3,2,1]$ 이 된다.

카드를 적절히 조작하여 $[1,2,3,4,5,6,7,8]$ 순서로 정렬하라.

조작 방법

- 카드를 클릭하면 해당 카드가 선택된다. 선택된 카드는 파란 테두리로 강조되며, 다시 클릭하면 선택이 해제된다.
- **◀ 2칸 앞으로** 버튼을 누르면 선택된 카드가 앞으로 2칸 이동한다. 카드가 1번째 또는 2번째 위치에 있으면 비활성화된다.
- **2칸 뒤로 ▶** 버튼을 누르면 선택된 카드가 뒤로 2칸 이동한다. 카드가 7번째 또는 8번째 위치에 있으면 비활성화된다.
- 화면 아래에 현재까지의 이동 횟수가 표시된다.
- **초기화** 버튼을 누르면 카드 배치와 이동 횟수가 초기화된다.

채점 기준

1. 이동 횟수와 관계없이 카드를 $[1,2,3,4,5,6,7,8]$ 순서로 정렬하면 전체 점수의 20%를 획득한다.
2. 가능한 최소 이동 횟수로 정렬하면 전체 점수의 100%를 획득한다.

카드를 한 번 클릭하면 선택되고, 다시 클릭하면 선택이 해제된다. 이동할 카드를 고른 뒤 **◀ 2칸 앞으로** 또는 **2칸 뒤로 ▶** 버튼을 누르면 된다. 카드가 맨 앞 두 칸 또는 맨 뒤 두 칸에 있으면 그 방향 버튼은 비활성화된다.

아래 순서대로, 그때그때 화면에 보이는 숫자 카드를 선택하고 버튼을 누르면 된다.

1. 8 선택 → **2칸 뒤로 ▶**
2. 8 선택 → **2칸 뒤로 ▶**
3. 7 선택 → **2칸 뒤로 ▶**
4. 4 선택 → **◀ 2칸 앞으로**
5. 6 선택 → **2칸 뒤로 ▶**
6. 8 선택 → **2칸 뒤로 ▶**
7. 7 선택 → **2칸 뒤로 ▶**
8. 6 선택 → **2칸 뒤로 ▶**
9. 5 선택 → **2칸 뒤로 ▶**
10. 4 선택 → **2칸 뒤로 ▶**
11. 1 선택 → **◀ 2칸 앞으로**
12. 1 선택 → **◀ 2칸 앞으로**
13. 1 선택 → **◀ 2칸 앞으로**
14. 3 선택 → **2칸 뒤로 ▶**

14번째 조작이 끝나면 카드가 $[1, 2, 3, 4, 5, 6, 7, 8]$ 순서가 된다. 역순 쌍(인버전)이란 배열에서 왼쪽 위치의 수가 오른쪽 위치의 수보다 큰 (왼쪽, 오른쪽) 쌍이다. 시작 순열 $[8, 7, \ldots, 1]$의 역순 쌍(인버전) 개수는 $\binom{8}{2} = 28$이고, 조작 한 번에 역순 쌍은 최대 2만 줄어들므로 $28 \div 2 = 14$가 이동 횟수의 하한이다. 위 순서는 이 하한에 맞춘 14회 조작이므로 만점이다. 정렬만 맞추면 부분점수 20%를 받을 수 있다.

위의 방법 이외에도 14회의 조작으로 카드를 $[1, 2, 3, 4, 5, 6, 7, 8]$ 순서로 정렬하면 정답 처리된다.

17. 로봇 (15점)

좌표평면 상에 로봇이 있다. 로봇은 좌표평면의 원점 $(0, 0)$에서 출발해 다음과 같은 세 가지 이동을 할 수 있다.

- $\rightarrow d$: 오른쪽으로 d 이동
- $\uparrow d$: 위로 d 이동
- $\downarrow d$: 아래로 d 이동

넓이는 초기에 0인 값이다. 로봇이 오른쪽 d만큼 이동할 때, 이동을 시작할 때의 y **좌표에** d**를 곱한 값**이 더해진다. $y < 0$이면 음수가 더해진다. 위/아래 이동은 넓이에 영향을 주지 않고, y 좌표에만 영향을 준다.

8장의 카드가 주어진다. 각 카드에는 여러 이동이 순서대로 적혀 있다. 카드를 사용하면 로봇은 해당 이동들을 순서대로 수행한다.

각 카드 **정확히 한 번씩** 사용해야 하며, 사용 순서는 자유롭게 정할 수 있다.

8장의 카드를 모두 사용하여, 넓이를 최대화하라.

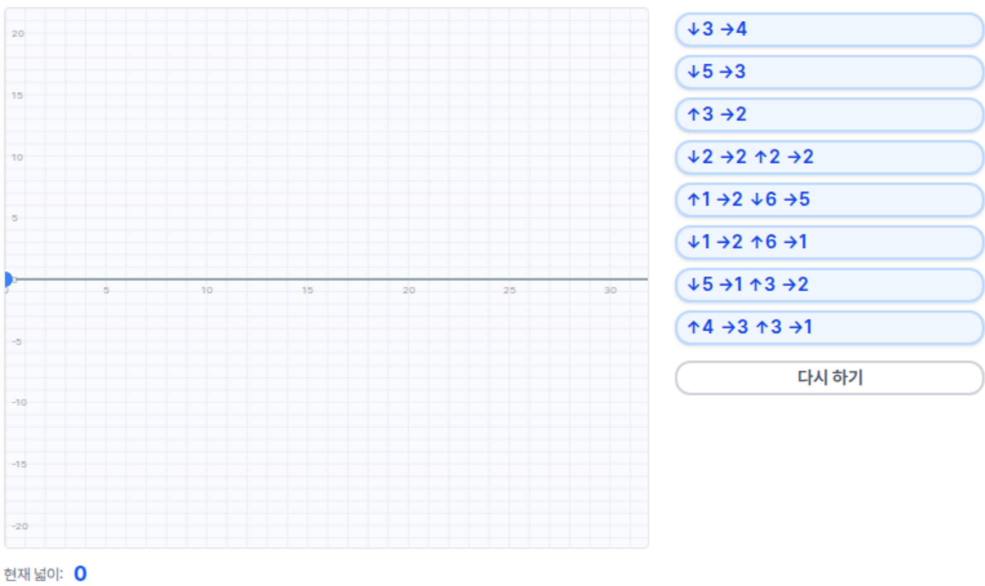
조작 방법

카드를 클릭하면 해당 카드에 적힌 이동이 순서대로 수행된다. 이미 사용한 카드는 다시 선택할 수 없다.

다시 하기 버튼을 누르면 모든 카드 사용이 초기화되고 로봇은 $(0, 0)$으로 돌아간다.

채점 기준

모든 카드를 사용했을 때 가능한 최대 넓이를 달성하면 전체 점수의 100%를 얻는다.



카드를 클릭하면 그 카드에 적힌 이동이 순서대로 실행된다. 각 카드는 정확히 한 번만 사용할 수 있으며, 이미 사용한 카드는 다시 선택할 수 없다.

만점을 위한 풀이는 **비율 정렬**이다. 각 카드마다 (위·아래 방향으로 움직이는 칸 수의 합)÷(오른쪽으로 움직이는 칸 수의 합)을 계산하고, 이 비율이 큰 카드부터 차례로 클릭하면 아래 순서가 나온다.

아래 순서대로 카드를 클릭하면 된다.

1. 「↑4 →3 ↑3 →1」이 적힌 카드를 클릭
2. 「↓1 →2 ↑6 →1」이 적힌 카드를 클릭
3. 「↑3 →2」가 적힌 카드를 클릭
4. 「↓2 →2 ↑2 →2」가 적힌 카드를 클릭
5. 「↓5 →1 ↑3 →2」가 적힌 카드를 클릭
6. 「↑1 →2 ↓6 →5」가 적힌 카드를 클릭
7. 「↓3 →4」가 적힌 카드를 클릭
8. 「↓5 →3」이 적힌 카드를 클릭

8번째 카드까지 실행이 끝나면 넓이가 253이 된다. 이는 가능한 최댓값이므로 만점이다.

18. 등비수열의 합 (15점)

당신은 상수 r에 대하여 식 $S = 1 + r + r^2 + \cdots + r^{2047} = 1 + \sum_{i=1}^{2047} r^i$의 값을 계산하는 프로그램을 작성하려고 한다. 당신은 세 변수 x, y, z를 사용할 수 있고, 각 변수의 초깃값은 모두 0이다.

당신은 다음과 같은 두 종류의 블록을 사용할 수 있다.

- 연산 블록: (변수) = (변수, 0, 1, r 중 하나) (+, × 중 하나) (변수, 0, 1, r 중 하나) 꼴의 연산을 할 수 있다. 등호의 왼쪽의 변수에 오른쪽의 연산값이 저장된다.
- 반복 블록: 0개 이상의 하위 블록을 여러 번 반복 실행할 수 있다. 반복 횟수는 상수로써 지정해야 한다.

연산 횟수는 실제로 실행되는 연산 블록의 총 개수로 정의된다.

- 연산 블록의 연산 횟수는 1회이다.
- 반복 횟수가 t이고 하위 블록들의 연산 횟수의 합이 k인 반복 블록의 연산 횟수는 $t \times k$회이다.
- 프로그램의 연산 횟수는 루트에 있는 모든 블록의 연산 횟수의 합이다.

당신의 목표는 50회 이하의 연산을 통해 x, y, z 중 하나의 변수의 값이 $1 + r + r^2 + \cdots + r^{2047}$이 되도록 하는 프로그램을 작성하는 것이다.

식 $S = 1+r+r^2+\cdots+r^{2047}$의 값과 x, y, z 중 하나의 변수의 값이 같다는 것은, 어떤 상수 r에 대해서도 두 값이 같다는 것이다. 예를 들어, $x = 1 + r$이라면, $r = 0$일 때 $x = 1$, $S = 1$로 일치하지만, $r = 1$일 때 $x = 2$, $S = 2048$로 일치하지 않으므로 답이 될 수 없다.

조작 방법

프로그램을 구성하는 편집 영역이 주어진다. 상단에는 목표식, 목표 달성 여부, 프로그램의 연산 횟수가 표시되어 있다.

"연산 블록 추가" 또는 "반복 블록 추가" 버튼을 눌러, 해당 블록을 루트 영역 또는 반복 블록 안에 추가할 수 있다.

또한, 연산 블록의 드롭다운을 눌러 피연산자 및 연산자를 수정하거나 버튼을 눌러 연산자를 수정할 수 있으며, 반복 블록에 수를 입력하여 반복 횟수를 수정할 수 있다.

각 블록의 "위로", "아래로", "삭제" 버튼을 눌러 블록들의 순서를 바꾸거나 제거할 수 있다.

블록들은 위에서 아래로 차례로 실행되며, 반복 블록 안에서는 해당 횟수만큼 반복하여 위에서 아래로 차례로 실행된다.

"되돌리기" 버튼을 눌러 직전 상태로 되돌리거나 "초기화" 버튼을 눌러 처음 상태로 되돌릴 수 있다.

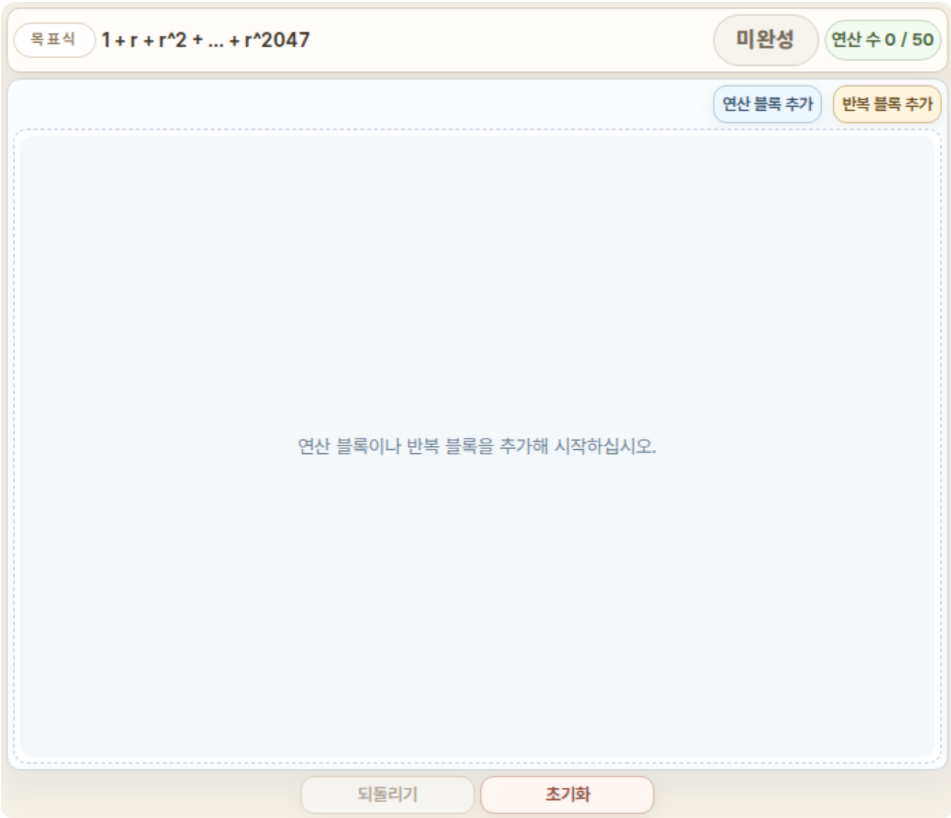
제약 조건

- 총 연산 횟수는 최대 50회이다.
  - 반복 블록을 사용할 수 있지만, 연산 횟수는 반복 블록을 모두 펼쳐 계산된다.
- 반복 블록은 최대 6중까지 중첩할 수 있다.

- 반복 블록에 입력 가능한 반복 횟수의 상한은 99회이다.

채점 기준

- 총 연산 횟수가 50회를 넘지 않고, 모든 연산을 실행한 후 x, y, z 중 하나의 값이 목표식과 정확히 일치해야 한다.



먼저 $x = r^2$, $y = 1 + r$로 만든다. 그 뒤 다음 세 연산을 10번 반복하면 된다.

$$z = x \times y, \qquad y = z + y, \qquad x = x \times x$$

반복을 시작할 때 $y = 1 + r + \cdots + r^{m-1}$, $x = r^m$이라고 하자. 그러면 $z = x \times y = r^m + r^{m+1} + \cdots + r^{2m-1}$가 된다. 따라서 $y = z + y$를 하면

$$y = 1 + r + \cdots + r^{2m-1}$$

가 된다. 마지막으로 $x = x \times x$를 하면 $x = r^{2m}$이 된다. 즉 한 번 반복할 때마다 y가 나타내는 항의 개수가 두 배가 된다.

처음에는 $m = 2$이고, 10번 반복하면 $m = 2048$이 된다. 따라서

$$y = 1 + r + r^2 + \cdots + r^{2047}$$

가 된다. 연산 횟수는 처음 2회와 반복 안의 3회씩 10번이므로 모두 32회이다. 이는 제한인 50회 이하이다.

위의 방법 이외에도 50회 이하의 연산으로 x, y, z 중 하나를 $1 + r + r^2 + \cdots + r^{2047}$로 만들면 정답 처리된다.

19. 간선 게임 (15점)

정점이 6개, 간선이 7개인 무방향 그래프가 있다. 그 중 두 정점을 각각 S, T라 한다.

당신과 상대는 번갈아 간선 하나와 1부터 7까지의 정수 중 하나를 골라, 그 정수를 그 간선에 배정한다. 이미 정수가 배정된 간선이나 이미 사용된 정수는 다시 고를 수 없다. 당신이 먼저 시작하며, 모든 간선에 정수가 배정되면 게임이 끝난다.

S에서 T로 가는 단순 경로의 **난이도**는 그 경로 위 간선에 배정된 정수 중 가장 큰 값이다. 게임이 끝났을 때, S에서 T로 가는 모든 단순 경로의 난이도 중 가장 작은 값을 이 그래프의 **난이도**라 한다.

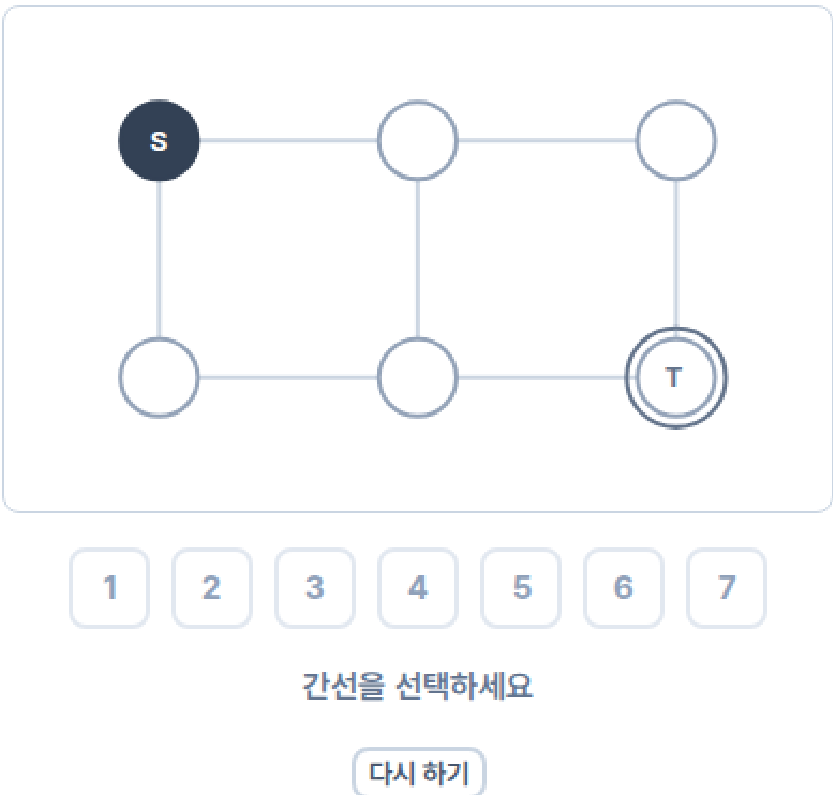
당신의 목표는 그래프의 난이도를 최소화하는 것이다. 상대는 난이도를 최대화하도록 최적으로 행동한다.

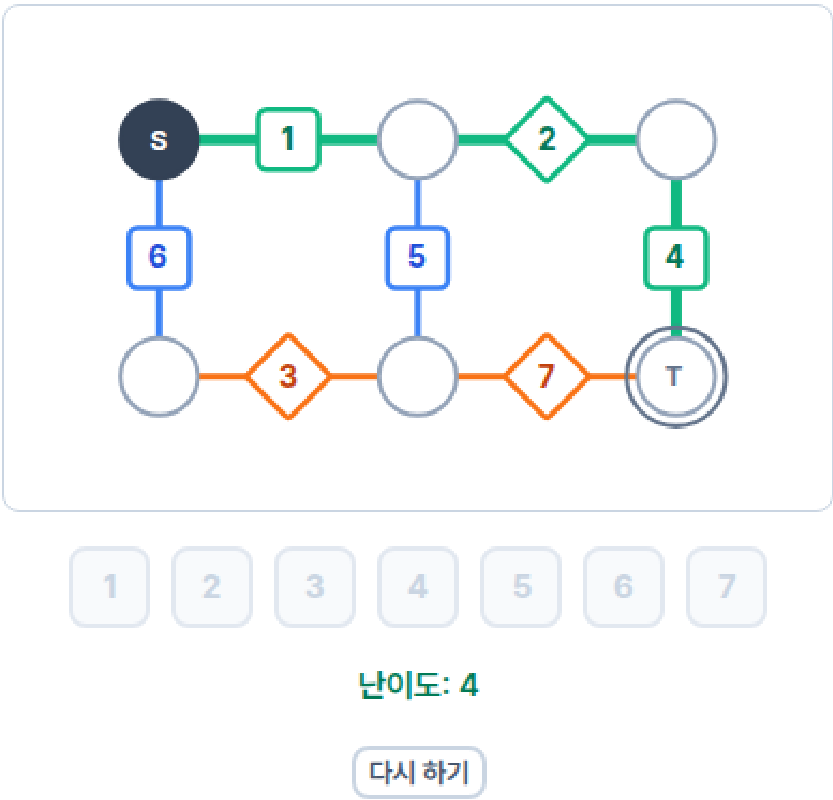
조작 방법

- 화면 아래에 1부터 7까지의 정수 버튼이 있다. 정수가 배정되지 않은 간선을 선택한 뒤 정수 버튼을 누르면, 해당 정수가 그 간선에 배정된다. 이미 정수가 배정된 간선은 다시 선택할 수 없다.
- 상대의 차례에는 상대가 자동으로 정수를 배정한다.
- 게임이 끝나면 그래프의 난이도가 표시되고, 그 값을 난이도로 갖는 경로 중 하나가 초록색으로 강조된다.
- “다시 하기” 버튼을 누르면 처음 상태로 돌아간다.

채점 기준

그래프의 난이도가 이론적으로 가능한 최솟값과 같으면 전체 점수의 100%를 받는다.





화면의 그래프에서 위쪽 세 간선으로 가는 경로와 아래쪽 세 간선으로 가는 경로를 생각하자. 두 경로의 가운데를 잇는 세로 간선에는 처음에 5를 배정한다.

그 다음에는 상대 차례가 끝날 때마다 화면을 보고 대응하면 된다. $1, 2, 3, 4$는 작은 수, $6, 7$은 큰 수로 생각한다. 위쪽 경로나 아래쪽 경로 중 한 곳에 큰 수가 놓이면 그 경로는 쓰지 않는다. 아직 다른 큰 수가 남아 있고 그 경로에 빈 간선이 있으면, 그 큰 수를 같은 경로의 빈 간선에 놓는다. 그 뒤에는 반대쪽 경로의 빈 간선들을 작은 수로 채우면 된다.

어느 경로에도 큰 수가 없다면 두 경로를 모두 살려 둔다. 작은 수만 놓인 경로 중 빈 간선이 하나뿐인 경로가 생기면, 그 빈 간선에 남은 작은 수를 놓아 그 경로를 완성한다. 그렇지 않으면 상대가 방금 작은 수를 놓은 경로의 반대쪽 경로에 남은 작은 수를 놓으면 된다.

이렇게 하면 게임이 끝났을 때 위쪽 경로나 아래쪽 경로 중 하나는 세 간선이 모두 $1, 2, 3, 4$ 중 하나로 채워진다. 처음에 가운데 세로 간선에 5를 놓았으므로, 난이도 4인 경로는 위쪽 경로나 아래쪽 경로에서 만들어진다. 이 그래프에서는 난이도 3 이하를 보장할 수 없으므로 난이도 4가 만점이다.

위의 방법 이외에도 마지막에 난이도가 4가 되도록 수를 배정하면 정답 처리된다.

20. 지연 탐색 (15점)

1 이상 2026 이하의 정수 x가 있다. 여러분은 x를 질문을 통하여 알아내어야 한다.

여러분은 x를 알아내기 위해 1 이상 2026 이하 정수인 y를 질문할 수 있다. 이후, 채점기는 x와 y의 대소 관계, 즉 $x < y$이거나, $x = y$이거나, $x > y$인지를 알려준다.

하지만, 채점기에는 결함이 있어 질문에 대한 답을 바로 알려주지 않고, <u>그 다음 질문이 들어온 후</u>에 직전 질문의 답을 알려준다.

여러분은 최소 횟수의 질문을 통하여 x를 알아내야 한다.

채점기는 고정된 하나의 x를 가지고 있지 않다. 초기화를 할 때 마다 채점기의 x가 바뀔 수 있다. 또한, 질문을 할 때 마다 이전 질문의 결과와 모순되지 않는 선에서 채점기는 들어온 모든 질문을 바탕으로 x를 변경하고, 이를 기반으로 직전 질문의 답을 결정할 수 있다.

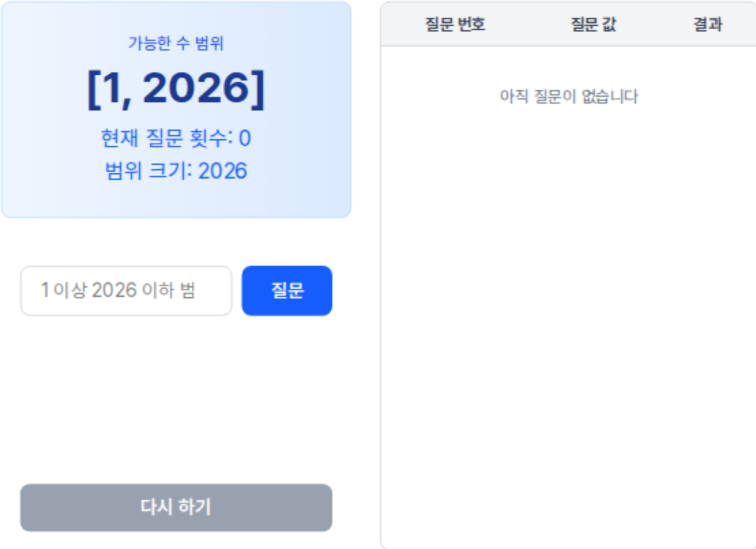
조작 방법

수 입력 창에 물어볼 y를 입력하고 질문 버튼을 누름으로써 질문을 할 수 있다. 다시 하기 버튼을 눌러 현재까지 진행된 모든 질문 내역을 초기화 할 수 있다.

채점 기준

여러분이 질문한 내용 중 채점기가 처음으로 $x = y$의 답을 반환한 질문이 i번째 질문이었다고 하자. i의 값에 따라 부분점수를 받을 수 있다. 첫 $x = y$ 응답 이후 추가로 이루어진 질문은 점수에 영향을 주지 않는다.

- $i \geq 25$인 경우 0%의 점수를 받을 수 있다
- $19 \leq i \leq 24$인 경우 30%의 점수를 받을 수 있다.
- $i = 18$인 경우 70%의 점수를 받을 수 있다.
- $i = 17$인 경우 80%의 점수를 받을 수 있다.
- $i \leq 16$인 경우 100%의 점수를 받을 수 있다.



채점기는 질문에 대한 답을 바로 알려주지 않는다. 다음 질문을 넣은 뒤에야 직전 질문의 답($x < y$, $x = y$, $x > y$)이 보인다. 그러므로 지금 보이는 답은 방금 넣은 질문의 답이 아니라, 그 바로 전 질문의 답이다.

D_k를 남은 질문이 k번일 때 확실히 처리할 수 있는 구간 길이라고 두면 다음을 얻는다.

$$D_1 = 1, \quad D_2 = 2, \quad D_k = D_{k-1} + 1 + D_{k-2} \quad (k \geq 3).$$

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D_k	1	2	4	7	12	20	33	54	88	143	232	376	609	986	1596	2583

가능한 x의 범위가 $[L, R]$이고 앞으로 k번 질문할 수 있을 때, 지금 물어볼 값은

$$y = L + D_{k-1}$$

로 잡는다. 만약 $y > R$이면 $y = R$을 묻는다. 답에 따라 L, R이 바뀌므로, 새 질문을 넣을 때마다 현재 가능한 범위의 왼쪽 끝 L을 기준으로 같은 식을 쓴다.

처음에는 $[L, R] = [1, 2026]$이고, 질문 16번 안에 맞히면 된다. 첫 질문은 $k = 16$에서 $1 + D_{15} = 1597$이다. 지연이 있으므로 1597을 물은 직후, 답이 보이기 전에 같은 $L = 1$에서 다음 값 $1 + D_{14} = 987$을 바로 이어서 입력한다. 이후에도 남은 질문 수가 줄어들 때마다 $L + D_{(\text{남은}-1)}$을 쓰고, 답이 보이면 그 답에 맞게 L, R을 바꾼다.

이 점화식은 지연 응답 때문에 생긴다. 남은 질문이 k번이고 지금 $y = L + D_{k-1}$을 물으면, 구간은 y보다 작은 쪽, y 하나, y보다 큰 쪽으로 나뉜다.

$x < y$이면 가능한 범위는 왼쪽 구간이다. 지연 때문에 y의 답을 보기 전에 이미 다음 질문을 하나 넣어 두었고, 그 질문은 왼쪽 구간을 기준으로 고른 값이다. 왼쪽으로 가는 경우에는 그 질문을 그대로 살려서 남은 $k - 1$번으로 길이 D_{k-1}까지 처리할 수 있다.

$x > y$이면 가능한 범위는 오른쪽 구간이다. 앞에서 넣어 둔 다음 질문은 왼쪽 구간을 기준으로 고른 값이므로 오른쪽에서는 쓸모가 없고, 질문 한 번을 낭비한 셈이 된다. 그래서 오른쪽으로 가는 경우에는 남은 $k - 2$번으로 길이 D_{k-2}까지 처리할 수 있다.

가운데의 y 하나를 맞히는 경우까지 합치면, 처리할 수 있는 길이는

$$D_{k-1} + 1 + D_{k-2}$$

가 된다. 그래서 $D_k = D_{k-1} + 1 + D_{k-2}$이다.

처음 가능한 범위 $[1, 2026]$의 길이는 2026이다. 표에서 $D_{16} = 2583$이고 $2583 \geq 2026$이므로, 위 규칙대로 질문하면 처음 $x = y$라는 답이 나오는 질문이 16번째 이내가 된다.

위의 방법 이외에도 지연 응답을 보면서 처음 $x = y$라는 답이 나오는 질문이 16번째 이내이면 정답 처리된다.